



Кафедра информатики, информационных систем и технологий

Петракова Н.В.

ОСНОВЫ HTML

Часть 1

*Учебно-методическое пособие
по дисциплине Web-программирование
для самостоятельной работы студентов
по направлению подготовки
09.03.03 Прикладная информатика*

`<html>`
`<title>`
`<head>`
`<meta>`
`<script>`
`<body>`
`<link>``<head>`

HTML

УДК 004.9 (07)
ББК 16.3
П 30

Петракова, Н. В. Основы HTML: учебно-методическое пособие по дисциплине «Web-программирование» для самостоятельной работы студентов по направлению подготовки 09.03.03 Прикладная информатика. Ч. 1 / Н. В. Петракова. – Брянск: Изд-во Брянский ГАУ, 2022. - 50 с.

В учебно-методическом пособии рассматриваются основные вопросы работы с HTML-документами: структурирование и форматирование текста, работа с графикой и мультимедиа, таблицами, гиперссылками, веб-формами и фреймами.

Учебно-методическое пособие предназначено для изучения дисциплины «Web-программирование» по направлению подготовки 09.03.03 Прикладная информатика. Кроме того, учебное пособие будет полезно студентам других специальностей и направлений, изучающих информационные технологии.

Рецензент: к.э.н., доцент кафедры информатики, информационных систем и технологий Федыкова Н.А.

Рекомендовано к изданию решением учебно-методической комиссии института энергетики и природопользования Брянского ГАУ, протокол №8 от 28 июня 2022 года.

© Брянский ГАУ, 2022
© Петракова Н.В., 2022

СОДЕРЖАНИЕ

Введение	4
Язык гипертекстовой разметки HTML: основные понятия и определения	5
Структура HTML-документа	7
Форматирование текста в HTML-документе	9
Спецсимволы HTML	13
Организация списков	14
Ссылки	17
Изображения	23
Таблицы	28
Формы	30
Фреймы	32
Задания для самостоятельной работы	35
Литература	38
Приложение	39

ВВЕДЕНИЕ

Развивается и широко используется мировая компьютерная сеть Интернет. Вместе с глобальным развитием сети Интернет в программировании все более отчетливо выделяется отдельная его отрасль – web-программирование. Необходимость в создании сайтов для различных компаний и организаций прямо пропорциональна уровню развития веб-технологий.

В учебно-методическом пособии «Основы HTML. Часть 1» рассмотрены: основные сведения о языке гипертекстовой разметки HTML, структура HTML-документов, разметка и форматирование текста на веб-страницы, работа с графикой, таблицами, гиперссылками, веб-формами и фреймами.

Материал сопровождается большим количеством примеров.

Также для закрепления теоретического материала в учебно-методическом пособии представлены задания для самостоятельной работы.

В результате освоения дисциплины обучающиеся должны:

1) знать:

- основы функционирования World Wide Web;
- язык гипертекстовой разметки HTML;
- технологию разделения содержимого и оформления с использованием каскадных таблиц стилей CSS;
- основные технологии web-программирования;

2) уметь:

- создавать статические HTML-страницы и применять таблицы стилей;
- кодировать на языках программирования;
- применять полученные знания для разработки web-сайтов;

3) владеть:

- практикой разработки базовых веб-приложений;
- навыками практического применения методов web-программирования.

ЯЗЫК ГИПЕРТЕКСТОВОЙ РАЗМЕТКИ HTML: ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

HTML (англ. *HyperText Markup Language* – язык гипертекстовой разметки) – это язык, принятый в WWW для создания и публикации web-страниц.

Гипертекст с помощью гиперссылок связывает между собой множество документов, которые называются web-документами, или web-страницами.

Разметка в HTML – это совокупность тегов, с помощью которых можно определить место на web-странице для картинки, текста, таблицы и др.

Язык – набор тегов, с помощью которых осуществляется разметка в HTML.

Основными конструкциями языка являются теги.

Тег – (англ. *tag* – именованная метка, дескриптор) – элемент языка разметки гипертекста.

Теги заключаются в угловые скобки и могут быть *парными* или *непарными (одиночными)*.

Любой тег состоит из 3 элементов: *открывающий тег (начальный)*, обозначаемого скобками $\langle \rangle$, *тело тега* (например, текстовой информации) и *закрывающий тег (конечный)* обозначаемого скобками \langle / \rangle .

Например:

```
<title>Моя первая web-страничка</title>
```

Пару, состоящую из открывающего и закрывающего тегов, называют **контейнером**.

Правильность написания парных тегов:

```
<тег1><тег2><тег3> ... </тег3></тег2></тег1>
```

Тег, который открывается первым должен закрыться последним, второй – предпоследним и т.д.

Теги могут иметь атрибуты, которые предоставляют дополнительную информацию об элементах HTML.

Атрибут – параметр или свойство элемента.

Атрибуты располагаются внутри начального тега и отделяются друг от друга пробелами. Значения атрибутов заключаются в кавычки (некоторые браузеры позволяют не использовать кавычки).

Атрибуты всегда используются в виде пары *имя/значение*.

Общий формат задания атрибутов имеет вид:

```
<имя_тега имя_атрибута="значение">
```

Например, тег:

<body bgcolor="red">

означает, что цвет фона страницы должен быть красным.

Например, тег:

<p align="center">

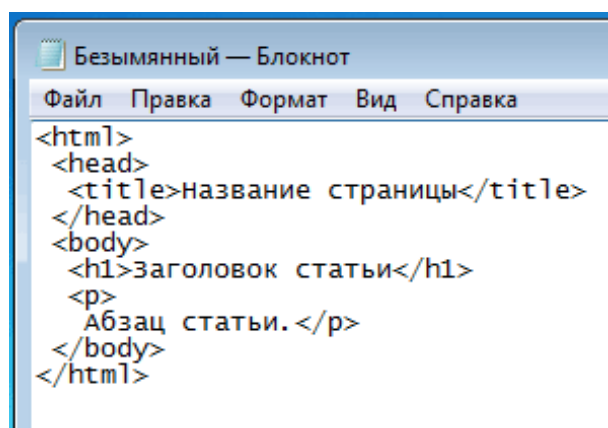
означает, что абзац необходимо выровнять по центру страницы отображения браузера.

Основы HTML содержат основные правила языка HTML, описание структуры HTML-документа между HTML-элементами (тегами).

HTML-документ – это обычный текстовый документ, может быть создан как в обычном текстовом редакторе (Блокнот), так и в специализированном, с подсветкой кода (Notepad++).

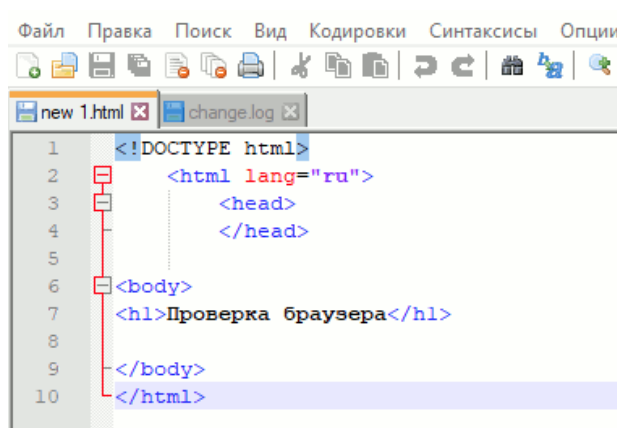
HTML-документ состоит из дерева HTML-элементов и текста.

HTML-документ имеет расширение .html.



```
<html>
<head>
<title>название страницы</title>
</head>
<body>
<h1>Заголовок статьи</h1>
<p>
Абзац статьи.</p>
</body>
</html>
```

Рис. 1. Блокнот



```
1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4 </head>
5
6 <body>
7 <h1>Проверка браузера</h1>
8
9 </body>
10 </html>
```

Рис. 2. Notepad++

Браузер просматривает (интерпретирует) HTML-документ, выстраивая его структуру и отображая ее в соответствии с инструкциями, включенными в этот файл. Если разметка правильная, то в окне браузера будет отображена HTML-страница, содержащая HTML-элементы – заголовки, таблицы, изображения и т.д.

Процесс интерпретации (парсинг) начинается прежде, чем веб-страница полностью загружена в браузер. Браузеры обрабатывают HTML-документы последовательно, с самого начала.

СТРУКТУРА HTML-ДОКУМЕНТА

HTML-документ состоит из двух частей:

- 1) заголовка документа, определяемого тегом **<head>**
- 2) тела документа, определяемого тегом **<body>**

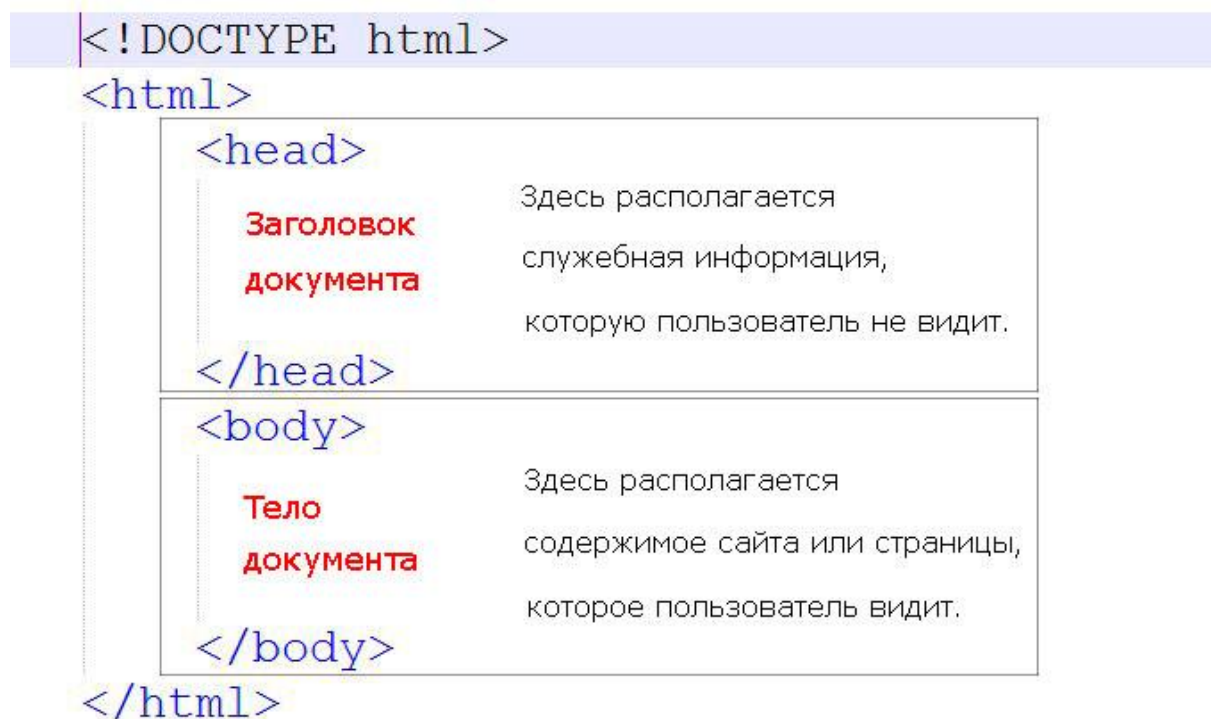


Рис. 3. Общая структура HTML-документа

Язык HTML следует правилам, которые содержатся в файле объявления типа документа (*Document Type Definition*, или *DTD*). DTD представляет собой XML-документ, определяющий, какие элементы, атрибуты и их значения действительны для конкретного типа HTML. Для каждой версии HTML есть свой DTD.

DOCTYPE отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в Интернете.

Элементы, находящиеся внутри элемента `<html>`, образуют дерево документа, так называемую объектную модель документа, DOM (*Document Object Model*). При этом элемент `<html>` является корневым элементом.

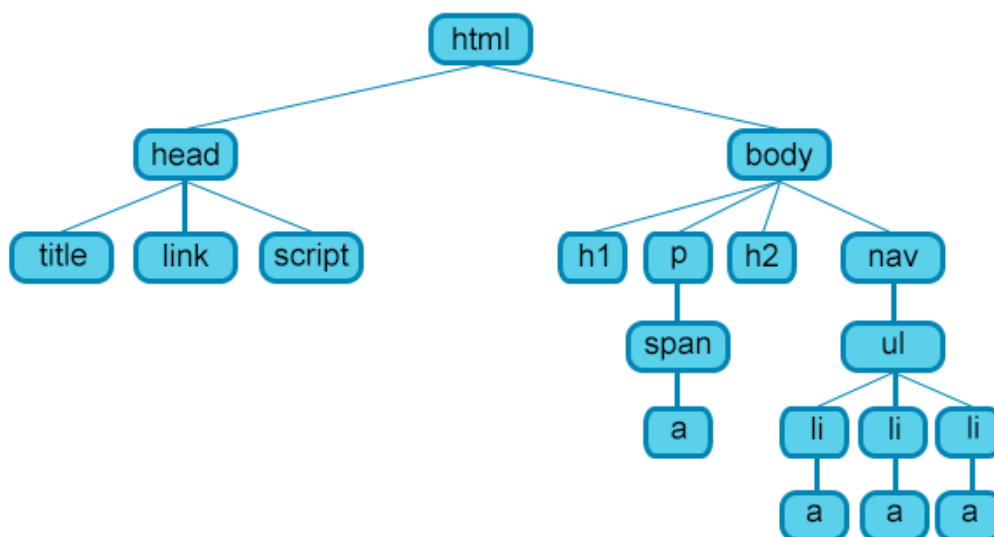


Рис. 4. Простейшая структура веб-страницы

Структурные теги

<html>

Является корневым элементом документа. Все остальные элементы содержатся внутри `<html>...</html>`. Все, что находится за пределами элемента, не воспринимается браузером как HTML-код и никак им не обрабатывается.

<head>

Раздел `<head>...</head>` содержит техническую информацию о странице: заголовок, описание, ключевые слова для поисковых машин, кодировку и т.д. Введенная в нем информация не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.

<title>

Обязательным элементом раздела `<head>` является `<title>`. Текст, размещенный внутри элемента `<title>`, отображается в строке заголовка веб-браузера. Длина заголовка должна быть не более 60 символов, чтобы полностью поместиться в заголовке. Текст заголовка должен содержать максимально полное описание содержимого веб-страницы.

<meta>

Необязательным элементом раздела `<head>` является элемент `<meta>`. Используется для хранения дополнительной информации о странице. Эту информацию используют браузеры для обработки страницы, а поисковые системы – для ее индексации. В блоке `<head>` может быть несколько элементов `<meta>`, так как в зависимости от используемых атрибутов они несут разную информацию.

<body>

В разделе `<body>` располагается все содержимое документа.

ФОРМАТИРОВАНИЕ ТЕКСТА В HTML-ДОКУМЕНТЕ



HTML-текст представлен в спецификации элементами для форматирования и группировки текста. Данные элементы являются контейнерами для текста и не имеют визуального отображения.

Элементы для форматирования текста несут смысловую нагрузку и обычно задают для текста, заключенного внутри, стилевое оформление.

Грамотно отформатированный текст дает понять поисковым системам, какие слова несут важную смысловую нагрузку, по каким из них предпочтительно ранжировать веб-страницу в поисковой выдаче. Вся текстовая информация, отображаемая на сайте, размещается внутри элемента **<body>**.

Параметры веб-страницы

Описание тегов тела документа следует начать с тега **<body>** который имеет атрибуты. Они задают параметры, распространяющиеся на весь документ, такие, как цвет фона, шрифта, гиперссылок и размеры полей страницы.

Цвета зависят от настроек браузера. Когда в HTML-программе цвета явно не заданы, браузер использует свои установки по умолчанию. Стандартные цвета Windows – черные буквы на белом фоне.

В теге **<body>** можно задать свои цветовые параметры документа, используя атрибуты:

bgcolor=цвет фона

text=цвет текста

Цвет можно задать просто, написав его название на английском языке или используя RGB-код, который состоит из # и шестизначного кода цвета (файл Таблица цветов).

Если в качестве атрибутов тега **<body>** указать:

<body bgcolor=green text= white>

или

<body bgcolor=#008000 text=#FFFAFA>

то цвет фона будет зеленым, текст будет белым.

В теге **<body>** можно задать также поля страницы.

При выводе Web-страницы на экран браузер автоматически оставляет небольшие поля (примерно 10 пикселей) вдоль левого и вдоль верхнего краев рабочего окна. Эти поля автоматически же заполняются указанным для данной страницы фоновым цветом и/или фоновым изображением, но никакие элемен-

ты страницы не могут заходить на эти поля. Если же требуется изменить размеры полей или убрать их совсем, нужно включить в тег `<body>` атрибуты **leftmargin** (*размер полей слева*) и **topmargin** (*размер полей сверху*).

В качестве значений для них после знака равенства указывается количество пикселей или ноль, чтобы отменить поля.

Например, тег `<body leftmargin=40>` создаст на всей странице левое поле шириной 40 пикселей. При n , равном 0, левое поле отсутствует.

Оформление текста

Заголовки

Заголовки являются важными элементами веб-страницы, они упорядочивают текст, формируя его визуальную структуру. Элементы `<h1>...<h6>` должны использоваться только для выделения заголовков нового раздела или подраздела.

Примеры использования заголовков:

Заголовок 1-го уровня

Заголовок 2-го уровня

Заголовок 3-го уровня

Заголовок 4-го уровня

Заголовок 5-го уровня

Заголовок 6-го уровня

Абзацы, средства переноса текста

Абзац формируется при помощи тега `<p> ...</p>`.

У тега `<p>` есть атрибут **align** (*выравнивание*), который может принимать значения:

- **left** – выровнять по левой границе;
- **right** – выровнять по правой границе;
- **center** – выровнять по центру;
- **justify** – выровнять по ширине.

По умолчанию абзац выравнивается по левой границе.

Все атрибуты располагаются в начальном теге:

`<p align="left"> </p>` – выравнивание абзаца по левому краю;

`<p align="center"> </p>` – выравнивание абзаца по центру;

`<p align="right"> </p>` – выравнивание абзаца по правому краю;
`<p align="justify"> </p>` – выравнивание абзаца по ширине страницы.

В качестве ограничителя абзацев используется *горизонтальная линейка*.

Тег `<hr>` – используется для разделения контента на веб-странице. Отображается в виде горизонтальной линии.

Тег `<hr>` может включать атрибуты:

size – определяет толщину линии в пикселях;

width – определяет размер линии в процентах от ширины экрана;

color – определяет цвет линии.

Пример:

```
<hr size=4 width=50% color=red>
```

Разрыв строки

Тег `
` – принудительный переход на новую строку.

Тег `
` полезен для разделения коротких фрагментов текста, таких как *постовые адреса* или *стихотворные строки*.

Форматирование текста

Стиль оформления текста

Существует несколько стандартных стилей оформления текста, для это используются следующие теги (табл. 1).

Таблица 1. Основные теги форматирования символов

Тег	Описание
<code>...</code>	полужирный текст
<code><i>...</i></code>	<i>курсив</i>
<code><s>...</s></code>	зачеркнутый текст
<code><u>...</u></code>	<u>подчеркнутый текст</u>
<code><sup>...</sup></code>	надстрочные символы (верхний индекс)
<code><sub>...</sub></code>	подстрочные символы (нижний индекс)
<code><tt>...</tt></code>	текст с фиксированной шириной символа (имитация пишущей машинки)
<code><big>...</big></code>	крупный шрифт
<code><small>...</small></code>	мелкий шрифт

Выбор шрифта

Для того, чтобы изменить цвет, размер и стиль шрифта для части текста, используют тег `...`.

Он имеет три основных атрибута:

- **size** – размер шрифта (принимает целые значения от 1 до 7);
- **color** – цвет текста;
- **face** – название шрифта.

HTML позволяет одновременно сочетать несколько тегов, предназначенных для изменения начертания шрифта, чтобы достичь особых эффектов.

Элементы анимации текста

Бегущая строка – теги `<marquee>...</marquee>`.

Регулируется время «пробегания» строки, цвет фона, ширина бегущей строки, цвет шрифта, вид его написания, наклон, размер.

Атрибуты для бегущей строки:

behavior="значение" – определяет тип движения текста и имеет значения:

- **alternate** – текст движется слева направо, достигнув края, меняет направление;
- **scroll** – бегущая строка, достигнув края, появляется с противоположной стороны;
- **slide** – бегущая строка останавливается у противоположного края страницы.

direction="значение" – определяет направление движения текста:

- **down** – определяет движение текста вниз;
- **up** – определяет движение текста вверх;
- **left** – бегущая строка движется справа налево;
- **right** – слева направо.

width – ширина бегущей строки (в пикселях или процентах).

height – высота бегущей строки (в пикселях или процентах).

bgcolor – цвет фона бегущей строки (цвет задается кодом или названием цвета).

scrolldelay – скорость прокрутки.

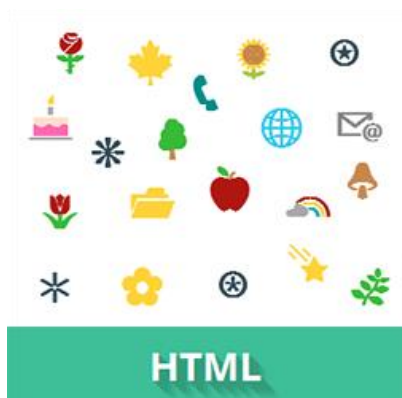
Комментарии

Комментарии в HTML имеют следующий синтаксис:

```
<!-- это комментарий -->
<!-- это тоже комментарий,
он занимает несколько строк -->
```

Информация в комментариях не имеет специального значения и никак не влияет на отображение документа.

СПЕЦСИМВОЛЫ HTML



Спецсимволы HTML, или символы-мнемоники, представляют собой конструкцию SGML (англ. **Standard Generalized Markup Language** – стандартный обобщённый язык разметки), ссылающуюся на определенные символы из символического набора документа. В основном они используются для указания символов, которых нет в стандартной компьютерной клавиатуре, либо которые не поддерживает кодировка HTML-страницы (Windows-1251, UTF-8 и т.д.).

Чтобы разместить символ на веб-странице, необходимо указать HTML-код или мнемонику.

Спецсимволы чувствительны к регистру, поэтому их необходимо прописывать точно так, как указано в таблице. Спецсимволы, не имеющие мнемоники, могут не отображаться вовсе или же некорректно отображаться в тех или иных браузерах.

Для вставки символов внутрь тегов необходимо воспользоваться HTML-кодом символа. Спецсимвол наследует цвет от цвета текста родительского элемента. Чтобы изменить цвет спецсимвола, можно поместить HTML-код внутрь тега `` или задать нужное значение свойства `color` (при вставке спецсимволов через свойство `content`).

Популярные спецсимволы HTML (Приложение 1)

1. [Полезные знаки и символы](#)
2. [Знаки пунктуации](#)
3. [Стрелки](#)
4. [Денежные знаки](#)
5. [Знаки зодиака](#)

ОРГАНИЗАЦИЯ СПИСКОВ



HTML-списки используются для группировки связанных между собой фрагментов информации.

Существует три вида списков:

- ✓ **маркированный список** – `` – каждый элемент списка `` отмечается маркером,
- ✓ **нумерованный список** – `` – каждый элемент списка `` отмечается цифрой,
- ✓ **список определений** – `<dl>` – состоит из пар термин `<dt>` – `<dd>` определение.

Маркированный список

Маркированный список представляет собой неупорядоченный список (от англ. *Unordered List*). Создается с помощью тега ``.

В качестве маркера элемента списка выступает метка.

Каждый элемент списка создается с помощью тега `` (от англ. *List Item*).

Для тега `` доступен атрибут:

`type="..."` – определяет формат маркера:

Виды маркера:

- 1) **disk** – диск (по умолчанию);
- 2) **circle** – окружность;
- 3) **square** – квадрат.

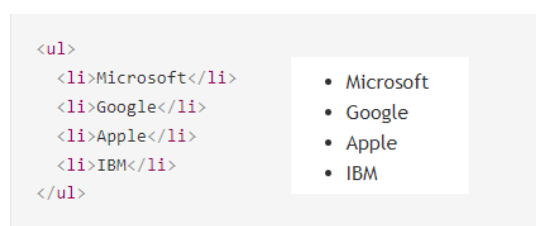


Рис. 5. Пример маркированного списка

Нумерованный список

Нумерованный список создается с помощью тега ``. Каждый пункт списка также создается с помощью тега ``. Браузер нумерует элементы по порядку автоматически и если удалить один или несколько элементов такого списка, то остальные номера будут автоматически пересчитаны.

Для тега `` доступен атрибут `value="n"`, который позволяет изменить номер по умолчанию для выбранного элемента списка.

Например, если для первого пункта списка задать `<li value="10">`, то остальная нумерация будет пересчитана относительно нового значения.

Для элемента доступны следующие атрибуты:

Таблица 2 - Атрибуты тега

Атрибут	Описание, принимаемое значение
reversed	Задаёт отображение списка в обратном порядке (например, 9, 8, 7...).
start	Задаёт начальное значение, от которого пойдёт отсчет нумерации, <i>например</i> , конструкция <ol start="10"> первому пункту присвоит порядковый номер «10». Также можно одновременно задавать тип нумерации, например, <ol type="I" start="10">.
type	Задаёт вид маркера для использования в списке (в виде букв или цифр). Принимаемые значения: 1 – значение по умолчанию, десятичная нумерация. A – нумерация списка в алфавитном порядке, заглавные буквы (A, B, C, D). a – нумерация списка в алфавитном порядке, строчные буквы (a, b, c, d). I – нумерация римскими заглавными цифрами (I, II, III, IV). i – нумерация римскими строчными цифрами (i, ii, iii, iv).

```
<ol>
  <li>Microsoft</li>
  <li>Google</li>
  <li>Apple</li>
  <li>IBM</li>
</ol>
```



Рис. 6. Пример нумерованного списка

Список определений

Списки определений создаются с помощью тега <dl>. Для добавления термина применяется тег <dt>, а для вставки определения – тег <dd>.

```
<dl>
  <dt>Режиссер:</dt>
  <dd>Петр Точилин</dd>
  <dt>В ролях:</dt>
  <dd>Андрей Гайдулян</dd>
  <dd>Алексей Гаврилов</dd>
  <dd>Виталий Гогунский</dd>
  <dd>Мария Кожевникова</dd>
</dl>
```

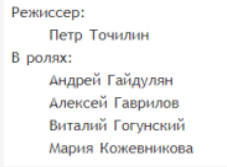


Рис. 7. Пример списка определений

Вложенный список

Зачастую возможностей простых списков не хватает, например, при создании оглавления никак не обойтись без вложенных пунктов.

Разметка для вложенного списка будет следующей:



Рис. 8. Пример вложенного списка

Многоуровневый нумерованный список

Многоуровневый список используется для отображения элементов списка на разных уровнях с различными отступами.

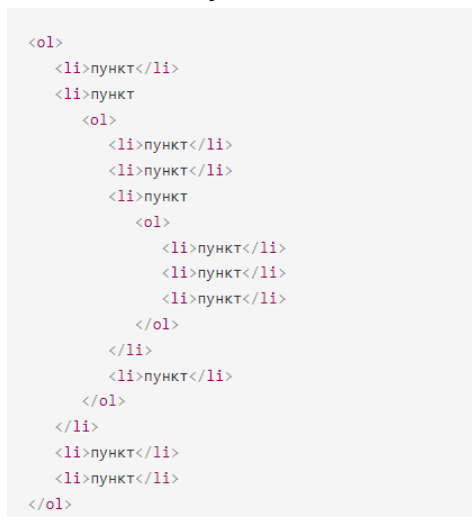


Рис. 9. Разметка для многоуровневого нумерованного списка

ССЫЛКИ



HTML-ссылки создаются с помощью тегов `<a>`, `<area>` и `<link>`. Ссылки представляют собой связь между двумя ресурсами, одним из которых является текущий документ.

Ссылки можно поделить на две категории:

- ✓ **ссылки на внешние ресурсы** – создаются с помощью элемента `<link>` и используются для расширения возможностей текущего документа при обработке браузером;
- ✓ **гиперссылки** – ссылки на другие ресурсы, которые пользователь может посетить или загрузить.

Структура ссылки

Гиперссылки создаются с помощью тега `<a>...`. Внутри помещается текст, который будет отображаться на веб-странице. Текст ссылки отображается в браузере с подчеркиванием, цвет шрифта – синий, при наведении на ссылку курсор мыши меняет вид.

Обязательным параметром тега `<a>` является атрибут **href**, который задает URL-адрес веб-страницы.

```
<a href="http://site.ru">указатель ссылки</a>
```

Ссылка состоит из двух частей – **указателя** и **адресной части**.

Указатель ссылки представляет собой фрагмент текста или изображение, видимые для пользователя.

Адресная часть ссылки пользователю не видна, она представляет собой адрес ресурса, к которому необходимо перейти.

Адресная часть ссылки состоит из URL.

URL (Uniform Resource Locator) – унифицированный адрес ресурса. При создании адресов для разделения слов между собой рекомендуется использовать дефис, а не символ подчеркивания. В общем виде URL имеющий следующий формат:

```
метод доступа://имя сервера:порт/путь
```

Метод доступа, или протокол, осуществляет обмен данными между рабочими станциями в разных сетях. Наиболее распространенные протоколы передачи данных:

file обеспечивает чтение файла с локального диска:

```
file:/gallery/pictures/summer.html
```

http предоставляет доступ к веб-странице по протоколу HTTP:

```
http://site.ru/
```

https – специальная реализация протокола HTTP, использующая шифрование (как правило, SSL или TLS):

```
https://site.ru/
```

ftp осуществляет запрос к FTP-серверу на получение файла:

```
ftp://pgu/directory/library
```

mailto запускает сеанс почтовой связи с указанным адресатом и хостом:

```
mailto: nika@gmail.com
```

Имя сервера описывает полное имя машины в сети, например, site.ru. Если имя сервера не указано, то ссылка считается локальной, т.е. она относится к той же машине, на которой находится HTML-документ, содержащий ссылку.

Номер порта TCP, на котором функционирует веб-сервер. Представляет собой число, которое необходимо указывать, если метод требует номер порта (отдельные сервера могут иметь свой отличительный номер порта). Если порт не указан, по умолчанию используется порт 80. Стандартными портами являются:

21 – FTP

23 – Telnet

70 – Gopher

80 – HTTP

Путь содержит имя папки, в которой находится файл.

Абсолютный и относительный путь

Когда в ссылке указывается только имя файла, браузер предполагает, что файл находится в той же папке, что и документ, содержащий гиперссылку. На практике веб-сайты содержат сотни документов, которые размещают в отдельные папки, чтобы ими было легче управлять. Чтобы создать ссылку на файл, находящийся вне папки, содержащей текущий документ, необходимо указать расположение файла или путь. HTML поддерживает два вида пути: *абсолютный* и *относительный*.

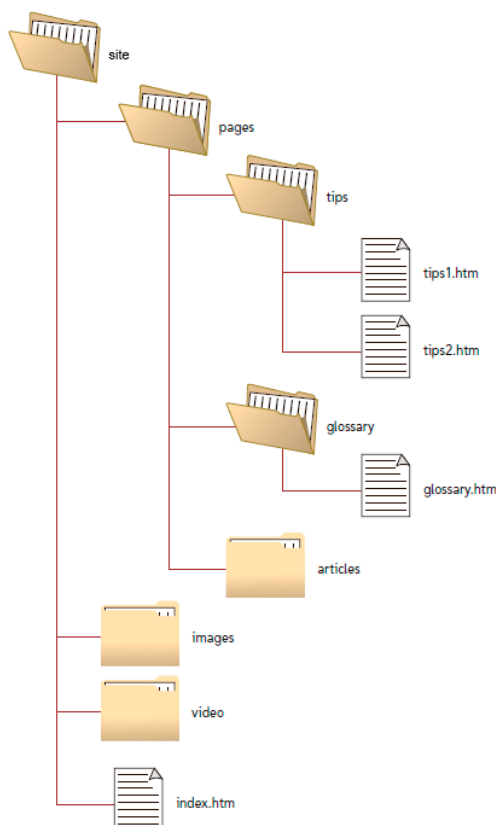


Рис. 10. Пример структуры папок

Абсолютный путь указывает точное местоположение файла в пределах всей структуры папок на компьютере (сервере). Абсолютный путь к файлу дает доступ к файлу со сторонних ресурсов и содержит следующие компоненты:

- протокол, например, `http` (опционально);
- домен (доменное имя или IP-адрес компьютера);
- папка (имя папки, указывающей путь к файлу);
- файл (имя файла).

Существует два вида записи абсолютного пути – с указанием протокола (полный) и без него:

```
http://site.ru/pages/tips/tips1.html
//site.ru/pages/tips/tips1.html
```

При ссылке на страницу на другом сайте, можно использовать только полный абсолютный путь.

Если файл находится в корневой папке, то путь к файлу будет следующим:

```
http://site.ru/index.html
```

При отсутствии имени файла будет загружаться веб-страница, которая задана по умолчанию в настройках веб-сервера (так называемый индексный файл).

```
http://site.ru/
```

Обычно в качестве индексного файла выступает документ с именем **index.html**. Наличие завершающего слеша / означает, что обращение идет к папке, если его нет – напрямую к файлу.

Относительный путь описывает путь к указанному документу относительно текущего. Путь определяется с учетом местоположения веб-страницы, на которой находится ссылка. Относительные ссылки используются при создании ссылок на другие документы на одном и том же сайте. Когда браузер не находит в ссылке протокол `http://`, он выполняет поиск указанного документа на том же сервере.

Относительный путь содержит следующие компоненты:

- папка (имя папки, указывающей путь к файлу);
- файл (имя файла).

Путь для относительных ссылок имеет три специальных обозначения:

- / указывает на корневую директорию и говорит о том, что нужно начать путь от корневого каталога документов и идти вниз до следующей папки;
- ./ указывает на текущую папку;
- ../ подняться на одну папку (директорию) выше.

Главное отличие относительного пути от абсолютного в том, что относительный путь не содержит имени корневой папки и родительских папок, что делает адрес короче, и в случае переезда с одного домена на другой не нужно прописывать новый абсолютный адрес. Но если сторонний ресурс будет ссылаться например, на ваши изображения с относительными адресами, то они не будут отображаться на другом сайте.

Якоря

Якоря, или внутренние ссылки, создают переходы на различные разделы текущей веб-страницы, позволяя быстро перемещаться между разделами. Это оказывается очень удобным в случае, когда на странице слишком много текста. Внутренние ссылки также создаются при помощи тега `<a>` с разницей в том, что атрибут **href** содержит имя указателя – так называемый якорь, а не URL-адрес. Перед именем указателя всегда ставится знак `#`.

Следующая разметка создаст оглавление с быстрыми переходами на соответствующие разделы:

```
<h1>Времена года</h1>
<h2>Оглавление</h2>
<a href="#p1">Лето</a> <!--создаём якорь, указав #id элемента-->
<a href="#p2">Осень</a>
<a href="#p3">Зима</a>
<a href="#p4">Весна</a>
<p id="p1">...</p> <!--добавляем соответствующий id элементу-->
<p id="p2">...</p>
<p id="p3">...</p>
<p id="p4">...</p>
```

Если нужно сделать ссылку с одной страницы сайта на определенный раздел другой страницы, то необходимо задать **id** для этого раздела страницы, а затем добавить его к абсолютному адресу ссылки:

```
<th id="about-color">color</th>
<a href="https://html5book.ru/css-shrifty/#about-color" class="site" target="_blank">color</a>
```

Как сделать изображение-ссылку

Чтобы сделать кликабельное изображение, необходимо поместить элемент `` внутрь элемента `<a>`. Чтобы ссылка открывалась в другом окне, нужно добавить атрибут **target="_blank"** для ссылки.

```
<a href="http://www.fast-torrent.ru/film/gran-za-granyu-tv.html" target="_blank"></a>
```

Как сделать ссылку на телефонный номер, скайп или адрес электронной почты

У ссылок появились новые возможности – по клику можно не только переходить на другие страницы и скачивать файлы, но и совершать звонки на телефоны, отправлять сообщения или звонить по скайпу.

ссылка на телефонный номер

```
<a href="tel:+74951234567">+7 (495) 123-45-67</a>
```

ссылка на адрес электронной почты

```
<a href="mailto:example@mail.ru">example@mail.ru</a>
```

ссылка на скайп (позвонить)

```
<a href="skype:имя-пользователя?call">Skype</a>
```

ссылка на скайп (открыть чат)

```
<a href="skype:имя-пользователя?chat">Skype</a>
```

ссылка на скайп (добавить в список контактов)

```
<a href="skype:имя-пользователя?add">Skype</a>
```

ссылка на скайп (отправить файл)

```
<a href="skype:имя-пользователя?sendfile">Skype</a>
```

ИЗОБРАЖЕНИЯ



HTML-изображения добавляются на веб-страницы с помощью тега ``. Использование графики делает веб-страницы визуально привлекательнее. Изображения помогают лучше передать суть и содержание веб-документа.

Теги `<map>` и `<area>` позволяют создавать карты-изображения с активными областями.

Тег ``

Тег `` представляет изображение и его резервный контент, который добавляется с помощью атрибута `alt`. Так как элемент `` является строчным, то рекомендуется располагать его внутри блочного элемента, например, `<p>` или `<div>`.

Для элемента `` доступны следующие атрибуты:

Таблица 3 - Основные атрибуты тега ``

Атрибут	Назначение
src	Задаёт путь к изображению. Синтаксис: src="flower.jpg"
alt	Добавляет альтернативный текст для изображения. Выводится на месте появления изображения до его загрузки или при отключенной графике, а также выводится всплывающей подсказкой при наведении курсора мыши на изображение. Синтаксис: alt="описание изображения"
height	Задаёт высоту элемента в пикселях. Синтаксис: height="300"
width	Задаёт ширину элемента в пикселях. Синтаксис: width="500"
usemap	Определяет изображение в качестве карты-изображения. Значение обязательно должно начинаться с символа <code>#</code> . Значение ассоциируется со значением атрибута name или id тега <code><map></code> и создаёт связь между элементами <code></code> и <code><map></code> . Атрибут нельзя использовать, если элемент <code><map></code> является потомком элемента <code><a></code> или <code><button></code> . Синтаксис: usemap="#mymap"

Адрес изображения

Адрес изображения может быть указан полностью (абсолютный URL), *например*: `url(http://anysite.ru/images/anyphoto.png)`

Или же через относительный путь от **документа** или **корневого каталога** сайта:

- `url(../images/anyphoto.png)` – относительный путь от документа;
- `url(/images/anyphoto.png)` – относительный путь от корневого каталога. Это интерпретируется следующим образом:
- `../` – означает подняться вверх на один уровень, к корневому каталогу;
- `images/` – перейти к папке с изображениями;
- `anyphoto.png` – указывает на файл изображения.

Размеры изображения

Без задания размеров изображения рисунок отображается на странице в реальном размере. Отредактировать размеры изображения можно с помощью атрибутов **width** и **height**. Если будет задан только один из атрибутов, то второй будет вычисляться автоматически для сохранения пропорций рисунка.

Тег <map>

Тег **<map>** служит для представления графического изображения в виде карты с активными областями. Активные области определяются по изменению вида курсора мыши при наведении. Щелкая мышью на активных областях, пользователь может переходить к связанным документам.

Для элемента доступен атрибут **name**, который задает имя карты. Значение атрибута **name** для тега **<map>** должно соответствовать имени в атрибуте **usemap** тега ****:

```

<map name="имя_карты" >
...
</map>
```

Тег **<map>** содержит ряд элементов **<area>**, определяющих интерактивные области в изображении карты.

Тег <area>

Тег **<area>** описывает только одну активную область в составе карты изображений на стороне клиента. Если одна активная область перекрывает другую, то будет реализована первая ссылка из списка областей.


```

<map name="имя_карты" >
<area атрибуты>
</map>

```

Таблица 4. Основные атрибуты тега <area>

Атрибут	Назначение
alt	Задаёт альтернативный текст для активной области карты.
shape	Задаёт форму активной области на карте и её координаты. Может принимать следующие значения: <ul style="list-style-type: none"> – default – стандартная форма – rect – прямоугольник – circle – круг – poly – многоугольник произвольной формы
coords	Определяет позицию области на экране. Координаты задаются в единицах длины и разделяются запятыми: для круга – координаты центра и радиус круга; для прямоугольника – координаты верхнего левого и правого нижнего углов; для многоугольника – координаты вершин многоугольника в нужном порядке, также рекомендуется указывать последние координаты, равные первым, для логического завершения фигуры.
href	Указывает URL-адрес для ссылки. Может быть указан абсолютный или относительный адрес ссылки.

Пример создания карты изображения

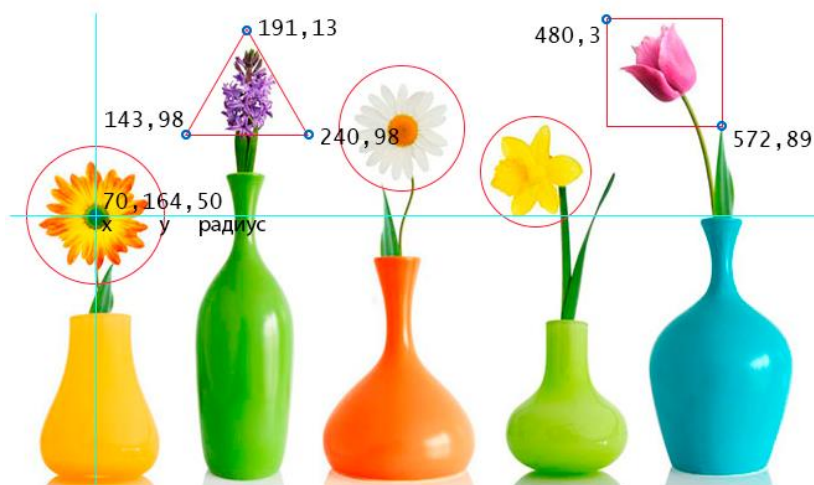


Рис. 11. Пример разметки изображения для создания карты

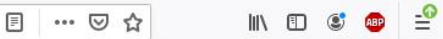
- 1) Размечаем исходное изображение на активные области нужной формы. Координаты областей можно вычислить с помощью программы для обработки фотографий, например, **Adobe Photoshop** или **Paint**.
- 2) Задаем имя карты, добавив его в тег **<map>** с помощью атрибута **name**. Это же значение присваиваем атрибуту **usemap** тега ****.
- 3) Добавляем ссылки на веб-страницы или части веб-документа для каждой активной области, по которым пользователь будет переходить при нажатии курсором мыши на активную область изображения.


```




<map name="flowers">
<area shape="circle" coords="70,164,50"
href="https://ru.wikipedia.org/wiki/Гербера" alt="gerbera" target="_blank">
<area shape="poly" coords="191,13,240,98,143,98,191,13"
href="https://ru.wikipedia.org/wiki/%C3%E8%E0%F6%E8%ED%F2" alt="hyacinth"
target="_blank">
<area shape="circle" coords="318,93,50"
href="https://ru.wikipedia.org/wiki/Помашка" alt="camomiles" tar-
get="_blank">
<area shape="circle" coords="425,129,45"
href="https://ru.wikipedia.org/wiki/Нарцисс_(растение)" alt="narcissus" tar-
get="_blank">
<area shape="rect" coords="480,3,572,89"
href="https://ru.wikipedia.org/wiki/Тюльпан" alt="tulip" target="_blank">
</map>
```



Рис. 12. Пример создания карты-изображения, при нажатии курсора мыши на цветок осуществляется переход на страницу с описанием

← → ↻ 🏠 <https://ru.wikipedia.org/wiki/Гербера>  Вы не представились системе [Обсуждение](#) [Вклад](#) [Создать учётную запись](#) [Войти](#)

Статья [Обсуждение](#) [Читать](#) [Править](#) [Править код](#) [История](#) 

 **Вики любит памятники: ваш взгляд на культурное наследие России!**  

Фотографируйте исторические и архитектурные памятники. Авторы лучших снимков получают призы!


Гербера

Материал из Википедии — свободной энциклопедии [[править](#) | [править код](#)]

Гербе́ра (лат. *Gerbera*) — род многолетних трав семейства **Астровые**, или **Сложноцветные** (*Asteraceae*). Имеется несколько десятков видов, большая часть которых растёт в Африке (в первую очередь в Южной Африке и на Мадагаскаре), некоторые виды встречаются также в тропической Азии. Цветки герберы по форме похожи на цветки представителей родов **Нивяник**, **Остеоспермум** и другие «ромашки»; окраска цветков герберы может быть любой, кроме синей.


Герберы выращивают во всём мире (в том числе в оранжереях) как красивоцветущие **декоративные растения** — в первую очередь на срезку, а также как **садовые растения**, иногда как комнатные растения.

Гербера



Цветки герберы различной окраски

Содержание [\[скрыть\]](#)


Википедия
 Свободная энциклопедия

[Заглавная страница](#)
[Рубрикация](#)
[Указатель А—Я](#)
[Избранные статьи](#)
[Случайная статья](#)
[Текущие события](#)

Участие

[Сообщить об ошибке](#)
[Сообщество](#)
[Форум](#)
[Свежие правки](#)
[Новые страницы](#)
[Справка](#)
[Пожертвовать](#)

Инструменты

[Ссылки сюда](#)
[Связанные правки](#)
[Служебные страницы](#)
[Постоянная ссылка](#)
[Сведения о странице](#)
[Цитировать страницу](#)

Рис. 13. Страница с описанием

ТАБЛИЦЫ

Меню ресторана "Ромашка"

Категория	Закусочные блюда		Первые блюда		Десерты
	Салаты	Закуски	Первые блюда	Вторые блюда	
Праздник	Венский салат	Винный салат	Суп с грибами	Вторые блюда	Пирожки
	Оливье	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
	Салат "Щетка"	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
Вкусные	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
Фрукты	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
Фрукты	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки
	Суп с грибами	Суп с грибами	Первые блюда	Вторые блюда	Пирожки

HTML

HTML-таблицы упорядочивают и выводят на экран данные с помощью строк или столбцов. Таблицы состоят из ячеек, образующихся при пересечении строк и столбцов.

Ячейки таблиц могут содержать любые HTML-элементы, такие как заголовки, списки, текст, изображения, элементы форм, а также другие таблицы. Каждой таблице можно добавить связанный с ней заголовок, расположив его перед таблицей или после нее.

Таблицы больше не используются для верстки веб-страниц и компоновки отдельных элементов, потому что такой прием не обеспечивает гибкость структуры и адаптивность сайта, существенно увеличивая HTML-разметку.

Создание таблицы

Таблица создается при помощи тега `<table>...</table>`, который является контейнером для элементов таблицы и все элементы должны находиться внутри него.

Например, с помощью данной разметки можно создать таблицу, состоящую из двух столбцов и двух строк:

```
<table>
<tr><th>текст заголовка</th><th>текст заголовка</th></tr> <!--ряд с ячейками заголовков-->
<tr><td>данные</td><td>данные</td></tr> <!--ряд с ячейками тела таблицы-->
</table>
```

текст заголовка	текст заголовка
данные	данные

Рис. 14. Внешний вид таблицы

Строки таблицы создаются с помощью элемента `<tr>`. Количество горизонтальных строк таблицы определяется количеством элементов `<tr>...</tr>`.

Тег `<th>` создает **заголовок столбца** – специальную ячейку, текст в которой выделяется полужирным. Количество ячеек заголовка определяется количеством элементов `<th>...</th>`.

Тег `<td>` создает ячейки таблицы, внутрь которых помещаются данные таблицы. Элементы `<td>...</td>`, расположенные в одном ряду, определяют количество ячеек в строке таблицы.

Количество пар ячеек `<td>` должно быть равно количеству пар ячеек `<th>`.

Для создания заголовка таблицы используется тег `<caption>`.

Таблица 5 - Основные атрибуты тега `<table>`

Атрибут	Назначение
align	Задаёт выравнивание таблицы. Допустимые значения: left – выравнивание таблицы по левому краю center – по центру right – по правому краю
border	Устанавливает толщину границы в пикселях вокруг таблицы.
bgcolor	Устанавливает цвет фона таблицы.
cellpadding	Определяет расстояние между границей ячейки и её содержимым. Этот атрибут добавляет пустое пространство к ячейке, увеличивая тем самым её размеры.
cellspacing	Задаёт расстояние между внешними границами ячеек.
width	Задаёт ширину таблицы в пикселях.

Таблица 6 - Основные атрибуты тега `<tr>` и `<td>`

Атрибут	Назначение
align	Задаёт горизонтальное выравнивание информации в ячейках. Допустимые значения: left – по левому краю center – по центру right – по правому краю justify – по ширине
valign	Вертикальное выравнивание информации в ячейках. Допустимые значения: top – выравнивание по верхнему краю строки middle – выравнивание по середине bottom – выравнивание по нижнему краю
colspan	Объединение ячеек по горизонтали (столбцов в строке).
rowspan	Объединение ячеек по вертикали (строк в столбце).
width	Задаёт ширину ячейки в пикселях или процентах.
nowrap	Запрещение перехода текста в ячейке на новую строку.

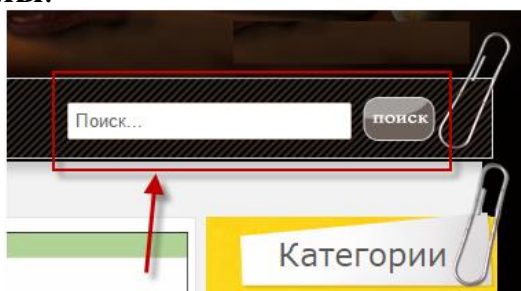
ФОРМЫ



Форма – это компонент веб-страницы с элементами управления, такими как текстовые поля, кнопки, флажки, диапазон или поле выбора цвета. Пользователь может взаимодействовать с такой формой, предоставляя данные, которые затем могут быть отправлены на сервер для дальнейшей обработки (например, возвращая результаты поиска или вычислений).

Примеры HTML-формы:

- форма поиска



- форма отправки комментариев

A screenshot of a comment form titled 'Добавить комментарий'. It contains three input fields: 'Имя (обязательно)', 'E-mail (не публикуется) (обязательно)', and 'Сайт'. Below these fields is an emoji picker with various emoji icons. At the bottom, there is a large text area for writing the comment and a yellow button labeled 'Отправить'.

- форма входа на сайт

A screenshot of a login form for 'Blog Good'. The form has a dark background with a red heart icon. It contains three input fields: 'Имя пользователя', 'Пароль', and a CAPTCHA field with the text '6 + 2 ='. Below the CAPTCHA field is a checkbox labeled 'Запомнить меня'. At the bottom, there is a blue button labeled 'Войти' and a link that says 'Забыли пароль?'.

- **форма обратной связи**

Форма обратной связи.

Ваше имя (обязательно)

Ваш E-Mail (обязательно)

Тема

Сообщение

- **форма для подачи заказа**

Заказать услугу

Ваше Имя

Ваш email

Ваш телефон

HTML-форма задается тегом `<form>...</form>`. Форма должна размещаться между тегами `<body>...</body>`.

Таблица 7 - Атрибуты тега `<form>`

Атрибут	Значение
name	Задаёт имя формы, которое будет использоваться для доступа к элементам формы.
action	Указывает путь к обработчику формы, является обязательным.
method	Задаёт способ передачи данных формы: <ul style="list-style-type: none"> – post – применяется для пересылки данных больших объёмов, а также конфиденциальной информации и паролей. Данные, отправляемые с помощью этого метода, не видны в заголовке URL, так как они содержатся в теле сообщения. – get (по умолчанию) – передаёт данные на сервер через адресную строку браузера. Этот метод нужно использовать, если не передаются большие объёмы информации.

Пользователи взаимодействуют с формами с помощью управляющих элементов, которые более подробно описаны в <https://site-do.ru/html/html11.php>.

ФРЕЙМЫ



Фреймы – это разделение окна браузера на отдельные области, в которых могут загружаться отдельные HTML-документы.

Для того, чтобы страница могла содержать фреймы, тег `<body>` заменяется на тег `<frameset>`. Тег `<frameset>` – это главный контейнер для создания фрейма, внутри которого содержатся другие элементы.

Структура HTML-документа с фреймами:

```
<html>
  <head>
    <title> фреймы </title>
  </head>
  <frameset>
  ...
  </frameset>
</html>
```

Таблица 8 - Атрибуты тега `<frameset>`

Атрибут	Значение	Пример
rows	Определяет расположение горизонтальных фреймов. Это разделенный запятыми список пикселей или процентов. По умолчанию используется 100%, что означает одну строку.	 <code><frameset rows="15%,15%,70"></code> или <code><frameset rows="15%,15%,*"></code>
cols	Определяет расположение вертикальных фреймов. Это разделенный запятыми список пикселей или процентов. По умолчанию используется 100%, что означает один столбец.	 <code><frameset cols="20%,20%,60"></code> или <code><frameset cols="20%,20%,*"></code>
border	Толщина границы между фреймами.	
bordercolor	Цвет линии границы.	
frameborder	Определяет, отображать рамку вокруг фрейма или нет.	
framespacing	Аналог атрибута <code>border</code> , задает ширину границы.	

Допускается использование вложенных друг в друга тегов **<frameset>** для создания сложной структуры фреймов. После задания структуры фреймов, каждый фрейм описывается при помощи тега **<frame>**. Закрывающий тег не требуется.

Таблица 9 - Атрибуты тега <frame>

Атрибут	Значение
name	Имя фрейма.
marginwidth	Отступ внутри фрейма по ширине.
marginheight	Отступ внутри фрейма по высоте.
src	Адрес веб страницы.
scrolling	Прокрутка фрейма: scrolling="yes" – всегда показывать полосу прокрутки scrolling="no" – никогда не показывать полосу прокрутки scrolling="auto" – показывать полосу прокрутки в том случае, если она необходима
noresize	Фиксированный размер фрейма.
frameborder	Задаёт ширину границы фрейма.
bordercolor	Задаёт цвет границы фрейма.

Некоторые браузеры не поддерживают фреймовую структуру документа или неправильно ее интерпретируют, кроме того зачастую пользователи в настройках своих браузеров умышленно отключают поддержку фреймовой структуры html документа.

Для того что бы дать понять пользователю, что его браузер/настройки браузера не поддерживают фреймы существует тег **<noframes>**.

Тег **<noframes>** выводит текст, заключенный в него в том случае, если браузер пользователя не поддерживает фреймы или они принудительно выключены в его настройках.

Ссылки во фреймах

В обычном HTML-документе при переходе по ссылке, в окне браузера текущий документ заменяется новым. При использовании фреймов схема загрузки документов отличается от стандартной. Основное отличие – возможность загружать документ в выбранный фрейм из другого. Для этой цели используется атрибут **target** тега **<a>**. В качестве значения используется имя фрейма, в который будет загружаться документ, указанный атрибутом **name**. Имя фрейма в атрибуте **name** должно быть уникальным.

Чтобы документ загружался в указанный фрейм, используется конструкция **target="имя фрейма"**.

Имя фрейма должно начинаться на цифру или латинскую букву.

Пример: `Щелкните здесь для загрузки документа menu.htm во фрейм с именем frame_1`

Плавающие фреймы

Плавающий фрейм – это фрейм, который можно добавлять в любое место веб-страницы. Еще одно его название – **встроенный фрейм**, он называется так из-за своей особенности встраиваться прямо в тело веб-страницы.

На рис. 15 приведен пример такого фрейма.

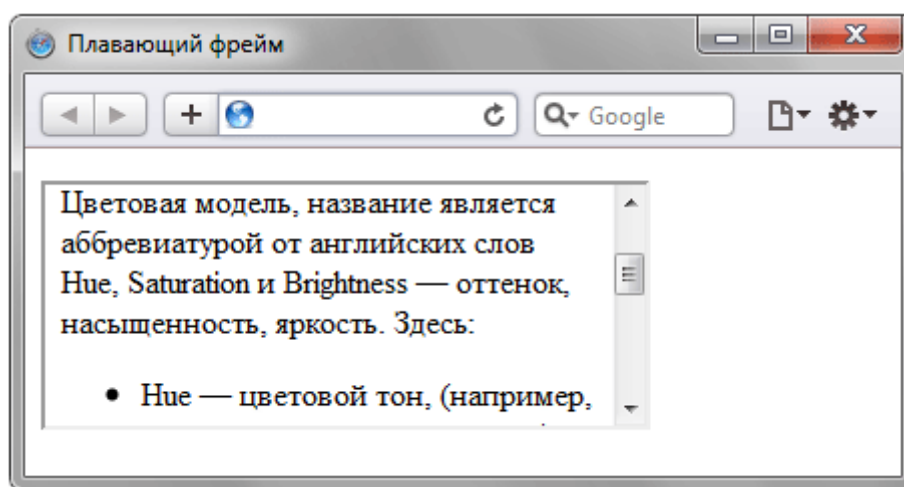


Рис. 15. Плавающий фрейм на веб-странице

Во фрейм можно загружать HTML-документ и прокручивать его содержимое независимо от остального материала на веб-странице. Размеры фрейма устанавливаются самостоятельно согласно дизайну сайта или собственных предпочтений.

Создание плавающего фрейма происходит с помощью тега `<iframe>`, он имеет обязательный атрибут **src**, указывающий на загружаемый во фрейм документ.

Ширина и высота фрейма устанавливается через атрибуты **width** и **height**. Если содержание не помещается целиком в отведенную область, появляются полосы прокрутки.

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задание 1.

1. Создать web-страницу с заголовком **Текст_Фамилия**.
2. Ввести текст «**Самостоятельная работа №1. Размещение и форматирование текста**», используя заголовки любого уровня.
3. Создать web-страницу (содержание произвольное, не менее 3 абзацев), для оформления применить теги управления внешним видом web-страницы, форматирования символов. В коде документа изменить **цвет фона и текста** (на усмотрение студента), **размеры полей** (произвольно). В конце страницы вставить символ авторского права и указать **Фамилию Имя Отчество**, институт, группу.
4. Разделить абзацы горизонтальными линиями с различными значениями атрибутов.

Задание 2.

1. Создать web-страницу с заголовком **Списки_Фамилия**.
2. Ввести текст «**Самостоятельная работа №2. Организация списков**», используя заголовки любого уровня.
3. Использовать элементы оформления внешнего вида HTML-документа (начертание, цвет шрифта, фон страницы и т.д).
4. Создать HTML-списки:

– **нумерованный:**

Интернет и World Wide Web

Здесь вы можете узнать:

- A. Что такое Интернет
- B. Историю создания Интернета
- C. Основные понятия среды Интернета

Интернет и World Wide Web

Здесь вы можете узнать:

- 5. Что такое Интернет
- 6. Историю создания Интернета
- 7. Основные понятия среды Интернета

– **список определений:**

Интернет и World Wide Web

Здесь вы можете узнать:

- Что такое Интернет
- Определение понятия
- Историю создания Интернета
- Кто придумал Интернет
- Основные понятия среды Интернета
- Составляющие Интернета

– **многоуровневый:**
Интернет и World Wide Web

Здесь вы можете узнать:

1. Что такое Интернет
 - Определенные понятия
2. Историю создания Интернета
 - Кто придумал Интернет
3. Основные понятия среды Интернета
Составляющие Интернета:
Электронная почта, телеконференции, поисковые системы ...

– **вложенный список:**

ОГЛАВЛЕНИЕ

- I. *Общие сведения о программировании.*
 1. *Введение.*
 2. *Что такое информатика?*
 3. *Что такое информация?*
 4. *Что такое вычислительная машина?*
 - *Общие положения.*
 - *Подробнее о памяти.*
 - *Ввод-вывод.*
 - *Оперативная память, внешняя память.*
 - *Порядок некоторых величин.*
 5. *Что может делать вычислительная машина?*
 6. *Что такое программирование?*
 7. *Несколько ключевых слов.*
 8. *Краткая история информатики.*
 - a. *Прединформатика.*
 - b. *Протоинформатика.*
 - c. *Информатика.*
 - d. *Библиография.*
 - e. *Задача.*
- II. *Введение в языки программирования.*

5. Создать бегущую текстовую строку (направление указать произвольно):
Фамилия Имя Отчество, Институт, Группа, Дата.

Задание 3.

1. Создать web-страницу с заголовком **Таблицы_Фамилия.**
2. Ввести текст «**Самостоятельная работа №3. HTML-таблицы**», используя заголовок любого уровня.
3. Создать таблицу с заголовком, включающую не менее 4 столбцов и 5 строк с текстовой, числовой информацией и специальными символами. Применить к таблице объединение ячеек, размер и выравнивание таблицы, использовать цвет и выравнивание текста в таблице.
4. Создать web-страницу с заголовком **Изображения_Фамилия.**
5. Ввести текст «**Самостоятельная работа №3. Ссылки. Изображения**», используя заголовок любого уровня.

6. Вставить изображение на web-страницу, сделать изображение-ссылку на страницу с таблицей.
7. Выполнить внутреннюю гиперссылку на странице с таблицей.

Задание 4.

1. Создать web-страницу с заголовком **Формы_Фамилия**.
2. Ввести текст «**Самостоятельная работа №3. HTML-формы**», используя заголовок любого уровня.
3. Создать форму по образцу, содержащую поля для ввода текстовой информации, список с возможностью выбора, радиопереключател, 2 кнопки, рамку оформления:

Форма для резюме специалиста

Кем вы хотите устроиться?

Какую зарплату вы хотите получать (в месяц)?
 100\$ 130\$

Задание 5.

Создать личный сайт на любую тему, отвечающий требованиям:

1. Продумать структуру сайта: число HTML-страниц, их содержимое;
2. Создать главную страницу сайта с помощью фреймов (не менее трех);
3. На главной странице должно быть навигационное меню для возможности перехода на любую другую страницу сайта;
4. Сайт должен иметь оригинальное оформление;
5. Стилистика оформления должна быть единой для всех страниц сайта;
6. У сайта должен быть логотип (фото), соответствующий тематике сайта и расположенный в шапке сайта, желательно использовать бегущую строку;
7. Добавить на сайт страницу с обратной связью, выполненной в виде HTML-формы.













ЛИТЕРАТУРА















1. Гумерова Л.З. Основы web-программирования: учебное пособие. Красноярск: Научно-инновационный центр, 2019. 104 с. Текст: электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: <https://www.iprbookshop.ru/97112.html>
2. Ефромеев Н.М., Ефромеева Е.В. Основы web-программирования: учебное пособие. Саратов: Вузовское образование, 2019. 128 с. Текст: электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. – URL: <https://www.iprbookshop.ru/86300.html>
3. Маркин А.В. Web-программирование: учебник. М.: Ай Пи Ар Медиа, 2021. 286 с. Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/104883.html>
4. Основы работы с HTML [Электронный ресурс]. 2-е изд. Электрон. текстовые данные. М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. 208 с. <http://www.iprbookshop.ru/73698.html>
5. Самоучитель HTML [Электронный ресурс]: <https://webref.ru/course/html-tutorial>
6. Основы Web-технологий [Электронный ресурс]: учебное пособие / П.Б. Храмцов и др. Электрон. текстовые данные. Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. 375 с. <http://www.iprbookshop.ru/67384.html>















ПРИЛОЖЕНИЕ

Приложение 1

Таблица 1 - Полезные знаки и символы

Вид	HTML-код	Описание
	🖂	Обратная сторона конверта
	☃	Снеговик
	⚓	Якорь
	✆	Знак телефона
	☎	Телефон
	☕	Горячие напитки
	✎	Карандаш, направленный вправо-вниз
	✏	Карандаш
	✐	Карандаш, направленный вправо-вверх
	✑	Незакрашенное острие пера
	✒	Закрашенное острие пера
	⚜	Геральдическая лилия

	❄	Снежинка
	❤	Закрашенное жирное сердце
	❅	Зажатая трилистниками снежинка
	❆	Жирная остроугольная снежинка
	★	Закрашенная звезда
	☆	Незакрашенная звезда
	✪	Незакрашенная звезда в закрашенном круге
	✫	Закрашенная звезда с незакрашенным кругом внутри
	✯	Вращающаяся звезда
	❉	Звёздочка с шарообразными окончаниями
	❋	Жирная восьмиконечная каплеобразная звездочка-пропеллер
	✲	Звёздочка с незакрашенным центром
	☀	Закрашенное солнце с лучами
	☁	Облака

	☂	Зонтик
	☑	Галочка в квадрате
	☒	Крестик в квадрате
	☹	Нахмуренный смайлик
	☺	Улыбающийся смайлик
	☻	Закрашенный улыбающийся смайлик
	☭	Серп и молот
	✿	Закрашенный цветок
	❀	Незакрашенный цветок
	✾	Цветок с шестью лепестками
	❁	Закрашенный обведенный цветок
	❂	Цветок из точек
	❦	Сердце в виде цветка
	✖	Жирный знак умножения











	✘	Жирный крестик
	✔	Жирная отметка галочкой
	✚	Жирный крест
	⚛	Символ атома
	❑	Незакрашенный квадрат с правой нижней тенью
	❒	Незакрашенный квадрат с правой верхней тенью
	◈	Алмаз в оправе
	◐	Круг с левой закрашенной половиной
	◑	Круг с закрашенной правой половиной
	⁂	Три звездочки

Таблица 2 - Знаки пунктуации

Вид	HTML-код	Описание
	 	Неразрывный пробел
<	<	Знак "меньше чем" (начало тега)
>	>	Знак "больше чем" (конец тега)

<<	&lquo;	Левая двойная угловая скобка
>>	&rquo;	Правая двойная угловая скобка
<	‹	Левая угловая одиночная кавычка
>	›	Правая угловая одиночная кавычка
"	"	Двойная кавычка
'	′	Одиночный штрих
”	″	Двойной штрих
‘	‘	Левая одиночная кавычка
’	’	Правая одиночная кавычка
‚	‚	Нижняя одиночная кавычка
“	“	Левая двойная кавычка
”	”	Правая двойная кавычка
„	„	Нижняя двойная кавычка
Ɔ	❜	Жирная одинарная верхняя запятая
ᄁ	❛	Жирная одинарная повернутая верхняя запятая

&	<code>&amp;</code>	Амперсанд
'	<code>&apos;</code>	Апостроф (одинарная кавычка)
§	<code>&sect;</code>	Параграф
©	<code>&copy;</code>	Знак соcopyright (авторское паво)
¬	<code>&not;</code>	Знак отрицания
®	<code>&reg;</code>	Знак зарегистрированной торговой марки
°	<code>&deg;</code>	Градус
±	<code>&plusmn;</code>	Плюс-минус
1	<code>&sup1;</code>	Верхний индекс "1"
2	<code>&sup2;</code>	Верхний индекс "2"
3	<code>&sup3;</code>	Верхний индекс "3"
1/4	<code>&frac14;</code>	Одна четверть
1/2	<code>&frac12;</code>	Одна вторая
3/4	<code>&frac34;</code>	Три четверти
'	<code>&acute;</code>	Знак ударения

μ	µ	Микро
¶	¶	Знак абзаца
•	·	Знак умножения
¿	¿	Перевернутый вопросительный знак
f	ƒ	Знак флорина
™	™	Знак торговой марки
•	•	Маркер списка
...	…	Многоточие
—	‾	Надчеркивание
-	–	Среднее тире
—	—	Длинное тире
}	}	Правая фигурная скобка
{	{	Левая фигурная скобка
=	=	Знак равенства
≠	≠	Знак неравенства

\cong	<code>&cong;</code>	Конгруэнтность (геометрическое равенство)
\approx	<code>&asymp;</code>	Почти равно
\leq	<code>&le;</code>	Меньше или равно
\geq	<code>&ge;</code>	Больше или равно
\sphericalangle	<code>&ang;</code>	Угол
\perp	<code>&perp;</code>	Перпендикулярно (кнопка вверх)
$\sqrt{\quad}$	<code>&radic;</code>	Квадратный корень
\sum	<code>&sum;</code>	N-ичное суммирование
\int	<code>&int;</code>	Интеграл
\otimes	<code>&#8251;</code>	Знак сноски
\div	<code>&divide;</code>	Знак деления
∞	<code>&infin;</code>	Знак бесконечности
@	<code>&#64;</code>	Символ собака
[<code>&#91;</code>	Левая квадратная скобка
]	<code>&#93;</code>	Правая квадратная скобка

Таблица 3 - Стрелки

Вид	HTML-код	Описание
←	←	Стрелка влево
↑	↑	Стрелка вверх
→	→	Стрелка вправо
↓	↓	Стрелка вниз
↔	↔	Стрелка влево-вправо
↙	↵	Стрелка вниз и влево - знак возврата каретки
⇐	⇐	Двойная стрелка налево
⇑	⇑	Двойная стрелка вверх
⇒	⇒	Двойная стрелка направо
⇓	⇓	Двойная стрелка вниз
↔	⇔	Двойная стрелка влево-вправо
➡	➠	Летящая стрела
➤	➤	Наконечник стрелы
↘	➥	Изогнутая стрела, указывающая вниз и вправо









	➦	Изогнутая стрела, указывающая вверх и вправо
	➳	Стрела направо
	↺	Круглая стрелка с наконечником против часовой стрелки
	↻	Круглая стрелка с наконечником по часовой стрелке
	⇧	Толстая полая стрелка вверх
	↩	Стрелка налево с крючком
	➫	Наклонённая вниз объёмная стрелка
	⬇	Закрашенная стрелка вниз
	⬆	Закрашенная стрелка вверх

Таблица 4 - Денежные знаки

Вид	HTML-код	Описание
¢	¢	Цент
£	£	Фунт стерлингов
₽	₽	Российский рубль
¥	¥	Йена или юань
€	€	Евро
\$	$	Доллар

₴	₴	Знак гривны
₹	₹	Индийская рупия
圓	圓	Китайский юань
₸	₸	Казахстанский тенге

Таблица 5 - Знаки зодиака

Вид	HTML-код	Описание
♈	♈	Овен
♉	♉	Телец
♊	♊	Близнецы
♋	♋	Рак
♌	♌	Лев
♍	♍	Дева
♎	♎	Весы
♏	♏	Скорпион
♐	♐	Стрелец
♑	♑	Козерог
♒	♒	Водолей
♓	♓	Рыбы

Учебное издание

Петракова Наталья Васильевна

ОСНОВЫ HTML

ЧАСТЬ 1

Учебно-методическое пособие
по дисциплине Web-программирование
для самостоятельной работы студентов
по направлению подготовки 09.03.03 Прикладная информатика

Редактор Осипова Е.Н.

Подписано к печати 13.10.2022 г. Формат 60x84¹/₁₆.
Бумага офсетная. Усл. п. л. 2,90. Тираж 25 экз. Изд. №7382.

Издательство Брянского государственного аграрного университета
243365 Брянская обл., Выгоничский район, с. Кокино, Брянский ГАУ