ПОДГОВТОВКА К ЭКЗАМЕНУ ТРИР

**1.** **Принципы адресации в Интернет.**

При подключении любого компьютера к сети Интернет он получает свой адрес, который используется в дальнейшем для идентификации этого компьютера. Интернет-адрес может состоять из двух частей: IP адреса и доменного адреса.

***IP адрес***имеет длину равную 32 бита. Для удобства он разделяется на четыре блока по 8 бит, которые записываются обычно как четыре десятичных числа (от 0 до 255), например, 192.45.9.200 (адрес сети - 192.45; адрес подсети – 9; адрес компьютера – 200). Это глобальная нумерация – адрес содержит полную информацию, необходимую для идентификации компьютера. Таким образом, каждый компьютер, подключенный к Интернету, имеет уникальный IP-адрес.

***Доменный адрес***определяет область, представляющую ряд хост-компьютеров. В отличие от цифрового адреса он читается в обратном порядке. Вначале идет имя компьютера, а затем имя сети, в которой он находится. Для возможности адресации были разработаны правила создания доменных адресов, однако особых правил, как следует называть домены, не существует. Тем не менее, в применении к доменам первого, самого верхнего уровня сложилась определенная практика. Так, в системе адресов Интернет приняты домены, представленные географическими регионами. Они имеют имя, состоящее из двух букв. Например, Франция – fr; Великобритания - uk; Канада – са; США – us; Германия – de; Россия – ru, Украина – ur, Казахстан – kz и т.д.

Существуют и домены, разделенные по тематическим признакам. Такие домены имеют трехбуквенное сокращенное название. Так, многие международные организации используют домены первого уровня com – для коммерческих, org и net – для некоммерческих организаций, gov – для правительственных учреждений, edu – для учебных заведений.

Служба, которая обеспечивает перевод имен компьютеров в их IP-

***адреса, называется Доменной Службой Имен (DNS - Domen Nanes System). Это что-то вроде гигантского, распределенного по многим компьютерам телефонного справочника, с IP-адресами вместо телефонов.***

*Имя компьютера*записывается как несколько слов, разделенных точками, например: *user12.tutor.ru.*Это отражает иерархическую, или доменную, структуру службы DNS. В данном случае *«User12»*- это имя компьютера в домене (второго уровня) *«tutor»*, принадлежащего домену (первого уровня) *«ru»*. Администратор, который отвечает за домен первого уровня *«ru»*(Россия), зарегистрировал домен второго уровня *«tutor.ru»*и передал туда все полномочия на регистрацию новых имен в пределах этого домена. В свою очередь администратор домена *«tutor.ru»*зарегистрировал имя «user12.tutor.ru» за определенным IP-адресом. Такая структура службы DNS обеспечивает, с одной стороны, уникальность имен компьютеров в пределах всего Интернета, а с другой стороны, четкое разделение административной ответственности (рис. 7.2).

Интернет-адресами для пользователей могут быть просто их регистрационные имена на компьютере, подключенном к сети. За именем следует знак @. Все это слева присоединяется к имени компьютера. Так пользователь, зарегистрировавшийся под именем *«vova»*на компьютере, имеющем в Интернет имя *«user12.tutor.edu»*,

будет иметь адрес:

*«vova@user12.tutor.edu».*

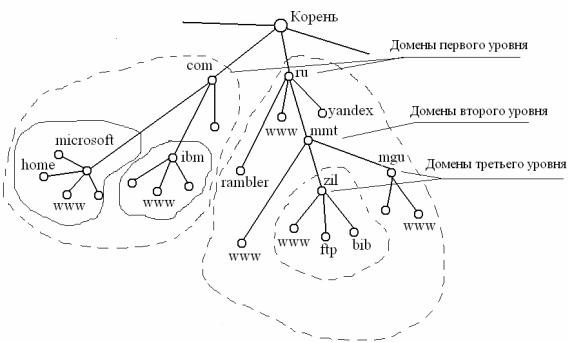
DNS – это особая служба Интернета, потому что она используется всеми остальными службами – от telnet до www. Перевод имен DNS в IPадреса происходит автоматически. Для этого надо только указать вашему компьютеру IP-адрес сервера DNS – того компьютера, которому будут направляться соответствующие запросы.

Так же, как каждый компьютер имеет свое уникальное имя, уникальное имя имеет и каждый документ в Интернете. Это уникальное имя назы-

***вается URL (Universal Resource Locator ) - Универсальный Указатель Ре-***

***сурса.***

URL имеет следующую форму:



*служба://имя\_компьютера/директория/поддиректория/.../имя\_файла*

*(например: http://roma.tutor.ru/people/peopl3.htm).*

***Служба***обозначается соответствующим протоколом, чаще всего http:// для веб-страниц и ftp:// для файловых архивов.

**2.** **Классы адресов в Internet.**

IP-адреса делятся на 5 классов (A, B, C, D, E). A, B и C — это классы коммерческой адресации. D – для многоадресных рассылок, а класс E – для экспериментов.

Класс А: 1.0.0.0 — 126.0.0.0, маска 255.0.0.0  
Класс В: 128.0.0.0 — 191.255.0.0, маска 255.255.0.0  
Класс С: 192.0.0.0 — 223.255.255.0, маска 255.255.255.0  
Класс D: 224.0.0.0 — 239.255.255.255, маска 255.255.255.255  
Класс Е: 240.0.0.0 — 247.255.255.255, маска 255.255.255.255

**3. Язык гипертекстовой разметки HTML. Синтаксис языка.**

1. Что такое HTML

TML, он же **HyperText Markup Languge** - язык разметки гипертекста, используется для описания структуры содержимого веб-страниц, а именно, расположения элементов друг относительно друга, их внешнего представлния, их функций. Элементами веб-страницы могут быть **заголовок, боковое меню, окошко поиска, странички с контентом, встроенное видео и т.д.**HTML Определяет, как они будут отображаться (шрифт, размер, используемые стили), как они будут расположены на странице, как они будут взаимодействовать (ссылки, выпадающие списки, зависимые поля). Для красивого отображения элементов в HTML и страницы в целом обычно используеттся **CSS (Cascading Style Sheets)** - документ описания стилей. Также в веб-страницы можно встраивать динамические элементы, например, браузерные игры, перемещаемые пользовалелем окошки, кнопки со сложным поведением и т.д. Все это становится возможным благодаря использованию скриптов, написанных на **JavaScript**, которые также явно указываются в HTML-документе.

Окончательный облик страницы формируется браузером после парсинга HTML-документа, описывающего эту страницу. В зависимости от браузера разные стили могут отображаться по-разному, а определенные элементы могут не поддерживатся некоторыми (обычно старыми или Explorer) браузерами. Все это надо учитывать при формировании HTML-документа.

2. Базовый синтаксис HTML

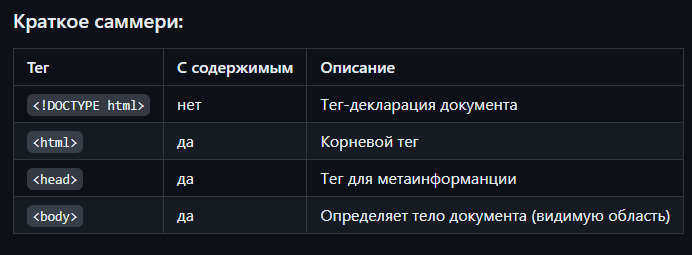
Обычно HTML-разметка страницы содержится в файле **index.html**.

HTML документ состоит из **тегов. HTML не чувствителен к регистру в имени тегов!**

Теги бывают **с содержимым: <head>Содержимое</head> и пустыми: <br>.** Теги можно вкладывать друг в друга.

Внутри тега можно задавать **параметры** или, что то же самое, **атрибуты**: **<section class="main">**

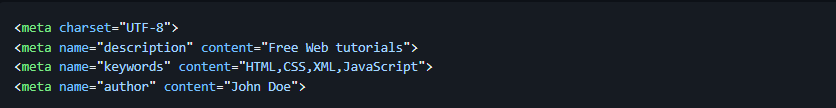
Каждый HTML документ начинается с тега-декларации **<!DOCTYPE html>**. Таким образом браузер понимает, что он читает именно HTML. Далее идет корневой тег , в который вложены все остальные теги - **<html></html>**. В нем находятся два тега **<head></head>** и **<body></body>.** Первый нужен для хранения метаинформации, которая не отображается на странице: заголовка, скриптов, стилей и т.д.. Второй используется для описания видимых частей страницы.



3. Тег **<head>**

Тег **<head>** используется для описания метаинформации о веб странице (информация об информации). Эта информация не видна пользователю, а используется браузерами и поисковыми движками.

Один из наиболее важных тегов, используемых в **<head>** это незакрывающийся **<meta>**. С его помощью можно задать используемую кодировку, добавить описание страницы, ключевые слова для поисковиков, имя автора



Внутри <head> можно задавать внутрениие стили для документа:

<style>

header {

text-allign: center;

}

</style>

И внешние стили:

<link rel="stylesheet" href="mystyle.css">

Также внутри <head> обычно указываются срипты:

<script>

function myFunction {

document.getElementById("demo").innerHTML = "Hello JavaScript!";

}

</script>

Еще один тег - <base> - используется для указания основного URL страницы, относительно которого разрешаются все остальные URL на странице:

<base href="https://www.w3schools.com/images/" target="\_blank">

4. HTML5 Layout

HTML5 предоставляет набор тегов для удобной разметки стандартной структуры большинства страниц

5. Форматирование текста

Текст в HTML обычно помещают в заголовках:

**<h1>**

Заголовок первого уровня

**</h1>**

**<h6>**

Заголовок шестого уровня

**</h6>**

в параграфах:

<p>

Какая-нибудь статья

**</p>**

**<pre>**

Какая-нибудь статья с сохранением переносов строк и пробелов

**</pre>**

в цитатах:

**<blockquote site="http://twitter.com">**

"Самое трудное в жизни - засунуть одеяло в пододеяльник" - Джейсон Стетхем

**</blockquote>**

и как код:

**<code>**

html,body {

padding: 0;

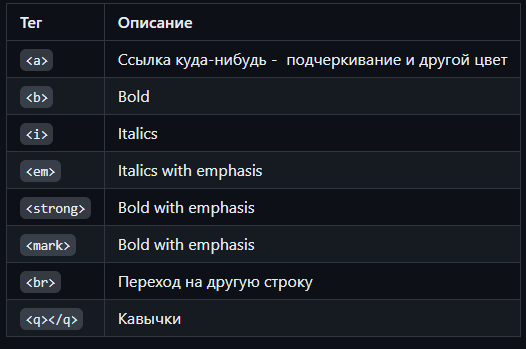
height: 100%;

margin: 0px;

overflow: hidden;

}**</code>**

Для форматирования используется следующий набор тегов:



Более детальное описание структуры HTML



**4. Основные элементы HTML страницы.**

Множество определяется в вопросе 3.

**5. Основные теги HTML для форматирования текста.**

**Теги <h1>-<h6>**

Теги [<h1>-<h6>](https://ru.w3docs.com/uchebnik-html/html-tegi-h1-h6.html) используются для структурирования [HTML заголовков](https://ru.w3docs.com/uchebnik-html/html-zagolovki.html). В HTML используются 6 уровней заголовков, от <h1> до <h6>. <h1> используется для обозначения самого важного, а <h6> наименее важного заголовков.

**Теги <b> и <strong>**

HTML теги [<b>](https://ru.w3docs.com/uchebnik-html/html-teg-b.html) и [<strong>](https://ru.w3docs.com/uchebnik-html/html-teg-strong.html) задают полужирное начертание шрифта. Разница между ними заключается в том, что тег <b> является тегом физической разметки, и выделяет текст без акцента на его важность. Тег <strong> же определяет текст, которому придают особую важность. Содержимое тега имеет большой вес для поисковиков, а устройства, считывающие с экрана, выделяют его определенной интонацией.

**Теги <i> и <em>**

Теги [<i>](https://ru.w3docs.com/uchebnik-html/html-teg-i.html) и [<em>](https://ru.w3docs.com/uchebnik-html/html-teg-em.html) задают курсивное начертание шрифта. Тег <i> текст является элементом физической разметки, то есть вложенный текст отличается только визуально и не воспринимается как важный браузерами и поисковыми машинами. Тег <em> экспрессивно-эмоционально выделяет фрагмент текста.

**Тег <pre>**

Тег [<pre>](https://ru.w3docs.com/uchebnik-html/html-teg-pre.html) используется для включения в HTML-документ предварительно отформатированного текста. Во вложенном в тег тексте сохраняются все пробелы и разрывы строк (как известно, браузеры по умолчанию любое количество идущих подряд пробелов показывают как один).

**Тег <mark>**

Тег [<mark>](https://ru.w3docs.com/uchebnik-html/html-teg-mark.html) определяет выделенный / подсвеченный текст. Визуально содержимое тега выглядит как отмеченное маркером желтого цвета.

**Тег <small>**

Тег [<small>](https://ru.w3docs.com/uchebnik-html/html-teg-small.html) определяет размер шрифта текста на один размер меньше, чем у родительского элемента. В HTML5 тег используется для хранения информации об авторских правах, а также определения мелкого, либо юридического шрифта.

**Тег <del> и <s>**

Тег [<del>](https://ru.w3docs.com/uchebnik-html/html-teg-del.html) выделяет часть текста, которая была удалена из документа.

Тег [<s>](https://ru.w3docs.com/uchebnik-html/html-teg-s.html) используется для определения текста, который больше не актуален.

**Тег <ins> и <u>**

Тег <ins> используется для определения части текста, которая была добавлена в документ. Содержимое тега в браузере отображается как подчеркнутый текст.

В тег [<u>](https://ru.w3docs.com/uchebnik-html/html-teg-u.html) заключается текст, который стилистически отличается от остального текста, например, слова с орфографическими ошибками или текст на другом языке.

**Теги <sub> и <sup>**

Тег [<sub>](https://ru.w3docs.com/uchebnik-html/html-teg-sub.html) используется для определения текста с нижним индексом. Тег выравнивает элемент как подстрочный. Тег [<sup>](https://ru.w3docs.com/uchebnik-html/html-teg-sup.html) используется для определения текста в верхнем индексе.

**Тег <dfn>**

Тег [<dfn>](https://ru.w3docs.com/uchebnik-html/html-teg-dfn.html) используется для выделения термина, который упоминается впервые. В браузере содержимое тега выделяется курсивом.

Теги <p>, <br> и <hr>

Тег [<p>](https://ru.w3docs.com/uchebnik-html/html-teg-p.html) определяет абзац в тексте. Друг от друга абзацы отделяются пустой строкой. Браузер автоматически добавляет верхний и нижний отступ, равный 1em, при этом отступы соседних абзацев «схлопываются».

Тег [<br>](https://ru.w3docs.com/uchebnik-html/html-teg-br.html) устанавливает перевод строки, то есть все написанное после него, будет перенесено на новую строку. В отличие от тега <p> перед строкой пустой отступ не добавляется.

В HTML5 тег [<hr>](https://ru.w3docs.com/uchebnik-html/html-teg-hr.html) используется для тематического разделения контента на странице. В предыдущих версиях HTML он использовался для создания горизонтальной линии на странице, визуально разделяя контент. В новой версии он приобрел смысловую нагрузку и определяет не только визуальный, но и тематический разрыв контента.

**6. Организация гиперссылок в HTML.**

## HTML-ссылки - Синтаксис

Тег HTML <a> определяет гиперссылку. Он имеет следующий синтаксис:

<a href="*url*">*текст ссылки*</a>

Наиболее важным атрибутом <a> элемента является href атрибут, который указывает место назначения ссылки.

Текст ссылки - это та часть, которая будет видна читателю.

Нажатие на текст ссылки отправит читателя по указанному URL-адресу.

По умолчанию ссылки будут отображаться следующим образом во всех браузерах:

* Ссылка, по которой никто не посещал, подчеркнута синим цветом
* Посещенная ссылка подчеркнута фиолетовым цветом
* Активная ссылка подчеркнута красным цветом

HTML-ссылки - целевой атрибут

По умолчанию связанная страница будет отображаться в текущем окне браузера. Чтобы изменить это, вы должны указать другой целевой объект для ссылки.

Атрибут target указывает, где открывать связанный документ.

Атрибут target может иметь одно из следующих значений:

* \_self - По умолчанию. Открывает документ в том же окне / вкладке, на которой был сделан щелчок
* \_blank - Открывает документ в новом окне или вкладке
* \_parent - Открывает документ в родительском фрейме
* \_top - Открывает документ во всей основной части окна

## Абсолютные URL-адреса по сравнению с Относительные URL-адреса

В обоих приведенных выше примерах используется **абсолютный URL** (полный веб-адрес) в href атрибуте.

Локальная ссылка (ссылка на страницу внутри того же веб-сайта) указывается с **относительным URL** (без "https://www" часть):

## HTML-ссылки - использовать изображение в качестве ссылки

Чтобы использовать изображение в качестве ссылки, просто поместите <img> тег внутри <a> тега:

## Ссылка на адрес электронной почты

Используйте атрибут mailto: внутри href для создания ссылки, которая открывает почтовую программу пользователя (чтобы позволить ему отправить новое электронное письмо).:

## Кнопка как ссылка

Чтобы использовать HTML-кнопку в качестве ссылки, вам необходимо добавить некоторый JavaScript-код.

JavaScript позволяет вам указать, что происходит при определенных событиях, таких как нажатие кнопки:

## Названия ссылок

Атрибут title указывает дополнительную информацию об элементе. Чаще всего информация отображается в виде текста всплывающей подсказки при наведении курсора мыши на элемент.

**6. Создание списков на Web-странице.**

**HTML-списки** используются для группировки связанных между собой фрагментов информации. Существует три вида списков:

* **маркированный список** — <ul> — каждый элемент списка <li> отмечается маркером,
* **нумерованный список** — <ol> — каждый элемент списка <li> отмечается цифрой,
* **список определений** — <dl> — состоит из пар термин <dt> — <dd> определение.

Каждый список представляет собой контейнер, внутри которого располагаются элементы списка или пары термин-определение.

Элементы списка ведут себя как блочные элементы, располагаясь друг под другом и занимая всю ширину блока-контейнера. Каждый элемент списка имеет дополнительный блок, расположенный сбоку, который не участвует в компоновке.

### 1. Маркированный список

**Маркированный список** представляет собой неупорядоченный список (от англ. Unordered List). Создаётся с помощью элемента <ul>. В качестве маркера элемента списка выступает метка, например, закрашенный кружок.

Браузеры по умолчанию добавляют следующее форматирование блоку списка:

ul {

padding-left: 40px;

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Каждый элемент списка создаётся с помощью элемента <li> (от англ. List Item).

Для элемента <ul> доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

### 2. Нумерованный список

**Нумерованный список** создаётся с помощью элемента <ol>. Каждый пункт списка также создаётся с помощью элемента <li>. Браузер нумерует элементы по порядку автоматически и если удалить один или несколько элементов такого списка, то остальные номера будут автоматически пересчитаны.

Блок списка также имеет стили браузера по умолчанию:

ol {

padding-left: 40px;

margin-top: 1em;

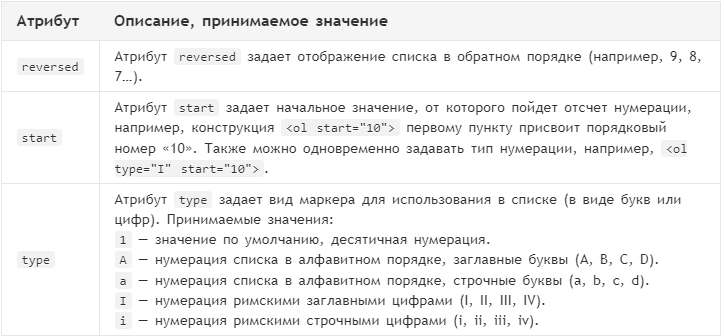
margin-bottom: 1em;

}

CSS

Для элемента <li> доступен атрибут value, 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 <ol> доступны следующие атрибуты:



### 3. Список определений

**Списки определений** создаются с помощью элемента <dl>. Для добавления термина применяется элемент <dt>, а для вставки определения — элемент <dd>.

Блок списка определений имеет следующие стили браузера по умолчанию:

dl {

margin-top: 1em;

margin-bottom: 1em;

}

CSS

Для элементов <dl>, <dt> и <dd> доступны [‎глобальные атрибуты](https://html5book.ru/html-attributes/).

### 4. Как создать вложенный список

Зачастую возможностей простых списков не хватает, например, при создании оглавления никак не обойтись без **вложенных пунктов**. Разметка для вложенного списка будет следующей:

<ul>

<li>Пункт 1.</li>

<li>Пункт 2.

<ul>

<li>Подпункт 2.1.</li>

<li>Подпункт 2.2.

<ul>

<li>Подпункт 2.2.1.</li>

<li>Подпункт 2.2.2.</li>

</ul>

</li>

<li>Подпункт 2.3.</li>

</ul>

</li>

<li>Пункт 3.</li>

</ul>

### 5. Многоуровневый нумерованный список

Многоуровневый список используется для отображения элементов списка на разных уровнях с различными отступами. Разметка для многоуровневого нумерованного списка будет следующей:

<ol>

<li>пункт</li>

<li>пункт

<ol>

<li>пункт</li>

<li>пункт</li>

<li>пункт

<ol>

<li>пункт</li>

<li>пункт</li>

<li>пункт</li>

</ol>

</li>

<li>пункт</li>

</ol>

</li>

<li>пункт</li>

<li>пункт</li>

</ol>

HTML

Такая разметка по умолчанию создаст для каждого вложенного списка новую нумерацию, начинающуюся с единицы. Чтобы сделать вложенную нумерацию, нужно использовать следующие свойства:

* counter-reset сбрасывает один или несколько счётчиков, задавая значение для сброса;
* counter-increment задаёт значение приращения счётчика, т.е. с каким шагом будет нумероваться каждый последующий пункт;
* content — генерируемое содержимое, в данном случае отвечает за вывод номера перед каждым пунктом списка.

**7. Работа с графическими изображениями в HTML.**

**HTML-изображения** добавляются на веб-страницы с помощью элемента <img>. Использование графики делает веб-страницы визуально привлекательнее. Изображения помогают лучше передать суть и содержание веб-документа.

Элементы <map> и <area> позволяют создавать **карты-изображения** с активными областями.

### 1. Элемент <img>

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

Элемент <img> имеет обязательный атрибут src, значением которого является абсолютный или относительный путь к изображению:

<img src="image.png" alt="Пример кода">

HTML

Для элемента <img> доступны следующие атрибуты:

|  |  |
| --- | --- |
| **Атрибут** | **Описание, принимаемое значение** |
| alt | Атрибут alt добавляет альтернативный текст для изображения. Выводится на месте появления изображения до его загрузки или при отключенной графике. Синтаксис: alt="описание изображения". |
| crossorigin | Атрибут crossorigin позволяет загружать изображения с ресурсов другого домена с помощью CORS-запросов. Изображения, загруженные в холст с помощью CORS-запросов, могут быть использованы повторно. Допускаемые значения: anonymous — Cross-origin запрос выполняется с помощью HTTP-заголовка, при этом учетные данные не передаются. Если сервер не даёт учетные данные серверу, с которого запрашивается контент, то изображение будет испорчено и его использование будет ограничено. use-credentials — Cross-origin запрос выполняется с передачей учетных данных. Синтаксис: crossorigin="anonymous". |
| height | Атрибут height задает высоту изображения в px. Синтаксис: height="300". |
| ismap | Атрибутismap указывает на то, что картинка является частью изображения-карты, расположенного на сервере (изображение-карта — изображение с интерактивными областями). При нажатии на изображение-карту координаты передаются на сервер в виде строки запроса URL-адреса. Атрибут ismap допускается только в случае, если элемент <img> является потомком элемента <a> с действительным атрибутом href. Синтаксис: ismap. |
| longdesc | URL расширенного описания изображения, дополняющее атрибут alt. Синтаксис: longdesc="http://www.example.com/description.txt". |
| src | Атрибут src задает путь к изображению. Синтаксис: src="flower.jpg". |
| sizes | Задаёт размер изображения в зависимости от параметров отображения. Работает только при заданном атрибуте srcset. Значением атрибута является одна или несколько строк, указанных через запятую. |
| srcset | Создаёт список источников для изображения, которые будут выбраны, исходя из разрешения экрана. Может использоваться вместе или вместо атрибута src. Значением атрибута является одна или несколько строк, разделенных запятой.  <img src="flower.jpg"  srcset="  img/flower-mobile.jpg 320w,  img/flower-wide-mobile.jpg 480w,  img/flower-tablet.jpg 768w,  img/flower-desktop.jpg 1024w,  img/flower-hires.jpg 1280w"  sizes="  (max-width: 20em) 30vw,  (max-width: 30em) 60vw,  (max-width: 40em) 90vw"  alt="Роза в моём саду">  HTML |
| usemap | Атрибут usemap определяет изображение в качестве карты-изображения. Значение обязательно должно начинаться с символа #. Значение ассоциируется со значением атрибута name или id элемента <map> и создает связь между элементами <img> и <map>. Атрибут нельзя использовать, если элемент <map> является потомком элемента <a> или <button>. Синтаксис: usemap="#mymap". |
| width | Атрибут width задает ширину изображения в px. Синтаксис: width="500". |

#### 1.1. 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.

#### 1.2. 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.

#### 1.3. Форматы графических файлов

* Формат **JPEG** (Joint Photographic Experts Group). Изображения JPEG идеальны для фотографий, они могут содержать миллионы различных цветов. Сжимают изображения лучше GIF, но текст и большие площади со сплошным цветом могут покрыться пятнами.
* Формат **GIF** (Graphics Interchange Format). Идеален для сжатия изображений, в которых есть области со сплошным цветом, например, логотипов. GIF-файлы позволяют установить один из цветов прозрачным, благодаря чему фон веб-страницы может проявляться через часть изображения. Также GIF-файлы могут включать в себя простую анимацию. GIF-изображения содержат всего лишь 256 оттенков, из-за чего изображения выглядят пятнистыми и нереалистичного цвета, как плакаты.
* Формат **PNG** (Portable Network Graphics). Включает в себя лучшие черты GIF- и JPEG-форматов. Содержит 256 цветов и дает возможность сделать один из цветов прозрачным, при этом сжимает изображения в меньший размер, чем GIF-файл.
* Формат **APNG** (Animated Portable Network Graphics). Формат изображения, основанный на формате PNG. Позволяет хранить анимацию, а также поддерживает прозрачность.
* **SVG** (Scalable Vector Graphics). SVG-рисунок состоит из набора геометрических фигур, описанных в формате XML: линия, эллипс, многоугольник и т.п. Поддерживается как статичная, так и анимированная графика. Набор функций включает в себя различные преобразования, альфа-маски, эффекты фильтров, возможность использования шаблонов. Изображения в формате SVG могут изменяться в размере без снижения качества.
* Формат **BMP** (Bitmap Picture). Представляет собой несжатое (оригинальное) растровое изображение, шаблоном которого является прямоугольная сетка пикселей. Bitmap-файл состоит из заголовка, палитры и графических данных. В заголовке хранится информация о графическом изображении и файле (глубина пикселей, высота, ширина и количество цветов). Палитра указывается только в тех Bitmap-файлах, которые содержат палитровые изображения (8 и менее бит) и состоят не более чем из 256 элементов. Графические данные представляют саму картинку. Глубина цвета в данном формате может быть 1, 2, 4, 8, 16, 24, 32, 48 бит на пиксель.
* Формат **ICO** (Windows icon). Формат хранения значков файлов в Microsoft Windows. Также, Windows icon, используется как иконка на сайтах в интернете. Именно картинка такого формата отображается рядом с адресом сайта или закладкой в браузере. Один ICO-файл содержит один или несколько значков, размер и цветность каждого из которых задаётся отдельно. Размер значка может быть любым, но наиболее употребимы квадратные значки со сторонами 16, 32 и 48 пикселей.

### 2. Элемент <map>

Элемент <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 атрибут name для элемента <map> должно соответствовать имени в атрибуте usemap элемента <img>:

<img src="url" usemap="#имя\_карты">

<map name="имя\_карты">

...

</map>

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

### 3. Элемент <area>

Элемент <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>

|  |  |
| --- | --- |
| **Атрибут** | **Краткое описание** |
| alt | Задает альтернативный текст для активной области карты. |
| 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. |
| download | Дополняет атрибут href и сообщает браузеру, что ресурс должен быть загружен в момент, когда пользователь щелкает по ссылке, вместо того, чтобы, например, предварительно открыть его (как PDF-файл). Задавая имя для атрибута, мы таким образом задаем имя загружаемому объекту. Разрешается использовать атрибут без указания его значения. |
| href | Указывает URL-адрес для ссылки. Может быть указан абсолютный или относительный адрес ссылки. |
| hreflang | Определяет язык связанного веб-документа. Используется только вместе с атрибутом href. Принимаемые значения — аббревиатура, состоящая из пары букв, обозначающих код языка. |
| media | Определяет, для каких типов устройств файл будет оптимизирован. Значением может быть любой медиа-запрос. |
| rel | Дополняет атрибут href информацией об отношении между текущим и связанным документом. Принимаемые значения: alternate — ссылка на альтернативную версию документа (например, печатную форму страницы, перевод или зеркало). author — ссылка на автора документа. bookmark — постоянный URL-адрес, используемый для закладок. help — ссылка на справку. license — ссылка на информацию об авторских правах на данный веб-документ. next/prev — указывает связь между отдельными URL. Благодаря этой разметке Google может определить, что содержание данных страниц связано в логической последовательности. nofollow — запрещает поисковой системе переходить по ссылкам на данной странице или по конкретной ссылке. noreferrer — указывает, что переходе по ссылке браузер не должен посылать заголовок HTTP-запроса (Referrer), в который записывается информация о том, с какой страницы пришел посетитель сайта. prefetch — указывает, что целевой документ должен быть кэширован, т.е. браузер в фоновом режиме загружает содержимое страницы к себе в кэш. search — указывает, что целевой документ содержит инструмент для поиска. tag — указывает ключевое слово для текущего документа. |
| shape | Задает форму активной области на карте и ее координаты. Может принимать следующие значения: rect — активная область прямоугольной формы; circle — активная область в форме круга; poly — активная область в форме многоугольника; default — активная область занимает всю площадь изображения. |
| target | Указывает, куда будет загружен документ при переходе по ссылке. Принимает следующие значения: \_self — страница загружается в текущее окно; \_blank — страница открывается в новом окне браузера; \_parent — страница загружается во фрейм-родитель; \_top — страница загружается в полное окно браузера. |
| type | Указывает MIME-тип файлов ссылки, т.е. расширение файла. |

### 4. Пример создания карты-изображения

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 ее в элемент <map> с помощью атрибута name. Это же значение присваиваем атрибуту usemap элемента <img>.
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.

<img src="https://html5book.ru/wp-content/uploads/2014/12/flowers\_foto.jpg" alt="flowers\_foto" width="680" height="383" usemap="#flowers">

<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/Ромашка" alt="camomiles" target="\_blank">

<area shape="circle" coords="425,129,45" href="https://ru.wikipedia.org/wiki/Нарцисс\_(растение)" alt="narcissus" target="\_blank">

<area shape="rect" coords="480,3,572,89" href="https://ru.wikipedia.org/wiki/Тюльпан" alt="tulip" target="\_blank">

</map>

**8. Использование фреймов на Web-страницах.**

**Frame (или "фрейм")** - это рамка, контейнер, область, в которую загружается сторонняя HTML-страница. Использование фраймов в HTML выглядит следующим образом:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Использование фреймов - "Нубекс"</title>

</head>

<frameset cols="200, \*">

<frame src="nubexMenu.html" name="nubexMenu">

<frame src="nubexContent.html" name="nubexContent">

</frameset>

</html>

Как видите, использование фреймов очень похоже на использование таблиц. В данном примере страница разбивается на две области (колонки), в одну из которых помещается меню (из стороннего файла html), а в другой располагается контент (из другого файла html). Такой принцип разработки сайта позволяет несколько снизить нагрузку на сайт, удобен для пользователя (посетитель работает с определенной, нужной областью страницы и содержимое страницы может меняться без ее перезагрузки: меняется только часть страницы - фрейм).

Фреймы применяются, например:

* При создании HTML-файлов справки;
* Если необходимо организовать загрузку файлов в определенную область страницы, при работе пользователя с другой областью.
* Для создания закрепленного блока (допустим, с контактами), который будет отображаться в определенной части, независимо от содержания других частей экрана.

Как видите, фреймы используются в определенных областях и заняли свою нишу (там, где их недостатки не имеют значения), но всё же использование фреймов на обыкновенных сайтах, которые стремятся привлекать поисковый трафик, приносит намного больше вреда, чем пользы. Давайте разберемся, почему фреймы понижают рейтинг сайта в глазах поисковых систем.

**Страницы с фреймами плохо индексируются**. Это большой и жирный минус, который перечеркивает всю пользу в использовании фреймов. Это связано с тем, что, по-сути, страница не имеет своего контента, а содержит множество ссылок на другие страницы. А страницы, которые содержат контент (т.е. те, которые подгружаются в определенные области сайта) будут индексироваться, но пользователь, перешедший на такую страницу из поиска, не получит полной информации - это будет всего лишь часть готовой страницы.

**9. Создание таблиц в HTML.**

[**https://html5book.ru/html-table/**](https://html5book.ru/html-table/)

**10. Формы. Организация форм на Web-странице.**

[**https://html5book.ru/formy/**](https://html5book.ru/formy/)

**11. Каскадные таблицы стилей.**

[**https://html5book.ru/osnovy-css/**](https://html5book.ru/osnovy-css/)

**12. Структура сайтов.**

[**https://neiros.ru/blog/marketing/struktura-sayta-vidy-algoritm-sozdaniya-i-11-servis-v-pomoshch/**](https://neiros.ru/blog/marketing/struktura-sayta-vidy-algoritm-sozdaniya-i-11-servis-v-pomoshch/)

**Выделенный хостинг (dedicated server).** Вы арендуете у хостера сервер целиком — то есть целый компьютер оказывается у вас в безраздельном пользовании. Вы можете устанавливать любое ПО и выбирать комплектующие. Вся мощность этой машины ваша.

Собственный сервер можно сделать и на базе обычного компьютера. Это потребует специфических знаний и затрат на обеспечение его работоспособности. Такой сервер тоже будет выделенным.

Преимущества выделенного хостинга:

* высокая производительность — на сервере проект только одного клиента;
* максимальный контроль — можно выбирать ПО и комплектующие, необходимые именно для вашего проекта.

Недостатки:

* дорого — стоимость аренды в среднем начинается от 100 долларов в месяц и может превышать 1000 долларов. Чем мощнее сервер, тем выше цена;
* для поддержки хостинга нужна высокая квалификация и большие затраты.

**Виртуальный хостинг (shared hosting).** Вид хостинга, когда один физический сервер сдают в одновременное пользование нескольким арендаторам. Арендаторов могут быть десятки и сотни. Специальное программное обеспечение следит за тем, чтобы мощности машины распределялись между клиентами более-менее равномерно, однако это происходит с некоторыми ограничениями.

Преимущества:

* лёгкий старт и простота управления;
* низкая стоимость — от 1,5 доллара в месяц.

Недостатки:

* Ограничены контроль и функциональность. Сервер предоставляют «как есть» и настраивают для всех клиентов одинаково. Нельзя, например, поставить дополнительное ПО или изменить файлы конфигурации.
* Ограничены объём и скорость передачи данных между пользователями и вашим сайтом.
* Могут быть проблемы с производительностью. Ресурсы сервера общие, клиенты используют их одновременно. Поэтому ваш сайт может начать виснуть из-за того, что в какой-то момент ему достанется мало оперативной памяти.
* Общий IP. На виртуальном хостинге под одним IP-адресом могут находиться сотни сайтов. И если одного из «соседей» заблокирует Роскомнадзор, то перестать работать могут и другие.

**Виртуальный сервер.** На одном физическом сервере запускают изолированные друг от друга процессы, которые имитируют работу самостоятельного компьютера. Виртуальные серверы обозначают аббревиатурами VPS (virtual private server) и VDS (virtual dedicated server). Технические различия между ними есть, но это несущественно для большинства пользователей.

Клиент получает те же привилегии, что и при работе с выделенным сервером. Это и установка необходимой операционной системы, и своё программное обеспечение. Виртуальные серверы не влияют на работу друг друга.

Преимущества:

* защищённая среда;
* контроль и возможность тонкой настройки;
* стоимость ниже, чем у выделенного сервера;
* скорость работы выше, чем у виртуального хостинга.

Недостатки:

* дороже виртуального хостинга;
* для администрирования необходимы знания.

**Облачный хостинг.** Он использует ресурсы нескольких связанных между собой физических и виртуальных серверов, к которым добавлены системы хранения данных. При аренде клиент платит только за фактически используемые ресурсы и может увеличивать или уменьшать их в зависимости от потребностей. Оплата может быть почасовой — это даёт возможность настроить потребление мощностей в дневное и ночное время. Настройки регулируются в автоматическом режиме, что обеспечивает гибкость управления.

Преимущества:

* стабильность за счёт динамического распределения ресурсов;
* экономичность;
* масштабируемость;
* гибкость.

Недостатки:

* дороже, чем виртуальный хостинг;
* меньше прав администрирования.