# МАТЕРИАЛ

## HTML

### Ссылки

Мы с Вами подошли к той теме, которая требует максимальное количество вашего внимания, поняв эту тему, вы без проблем сможете выстроить структуру будущего сайта.

Основная функция ссылок заключается в том, что они позволяют переходить с одного HTML документа на другой, либо содержать адрес файла, предназначенного для открытия в браузере пользователя, или для его дальнейшего скачивания.

Если вы активно пользуетесь интернетом, то, думаю, вы замечали, что ссылки могут ссылаться на страницу / файл, который находится совершенно на другом сервере, давайте рассмотрим от чего это зависит.

В HTML гиперссылки (или просто "ссылки") определяются тегом <a> (HTML Anchor Element).

Например:

| <!DOCTYPE html> <html>  <head>  <title>Пример размещения ссылки в HTML документе<title>  </head>  <body>  <a href="http://www.yandex.ru">Найдется всё</a>  </body> </html> |
| --- |

В браузере же это будет отображаться по умолчанию следующим образом: Найдется всё.

Как мы с вами знаем, некоторые теги используются только совместно с атрибутами (бесполезны сами по себе) и это снова тот случай. Атрибут href (аббревиатура от hyper reference) - это основной атрибут тега <a>, указывающий браузеру, URL-адрес страницы, куда ведет ссылка. Единый указатель ресурса (англ. Uniform Resource Locator, URL) — единообразный локатор (определитель местонахождения) ресурса.

#### Ссылки внутри страницы

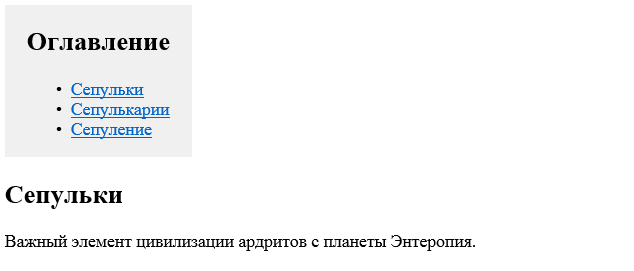
Ссылки допустимо указывать не только на другой сайт или документ, но и на элемент внутри страницы. Такая возможность применяется для добавления оглавления страницы с быстрым переходом к нужному разделу (так делает, например, Википедия) или для создания различных элементов интерфейса вроде вкладок.

Элемент, на который требуется сделать ссылку, обозначается идентификатором, а адрес ссылки меняет свой вид на #id, как показано в примере:

| <!DOCTYPE html> <html> <head>  <title>Пример размещения внутристраничной ссылки</title> </head> <body>  <div style="margin-bottom: 400px">  <h2>Оглавление</h2>  <ul>  <li><a href="#t1">Сепульки</a></li>  <li><a href="#t2">Сепулькарии</a></li>  <li><a href="#t3">Сепуление</a></li>  </ul>  </div>  <h2 id="t1">Сепульки</h2>  <p style="margin-bottom: 400px">Важный элемент цивилизации ардритов с планеты Энтеропия.</p>  <h2 id="t2">Сепулькарии</h2>  <p style="margin-bottom: 400px">Устройства для сепуления.</p>  <h2 id="t3">Сепуление</h2>  <p style="margin-bottom: 400px">Занятие ардритов с планеты Энтеропия.</p> </body> </html> |
| --- |

Здесь мы были вынуждены использовать CSS, чтобы получить эффект пролистывания страницы. Позже мы разберем его подробнее, а пока просто добавьте атрибуты “как есть”.

Результат данного примера показан ниже. К каждому заголовку <h2> добавлен уникальный идентификатор через атрибут id, а сама ссылка имеет вид #id, где id — идентификатор элемента на который делается переход.



Как видите, по своему виду такие ссылки ничем не отличаются от ссылок на другой сайт. Но стоит открыть любую ссылку в примере, как браузер переместит фокус к заголовку, на который ведёт ссылка. При этом немного поменяется и адресная строка, в конец пути будет добавлено #t1 или подобное. Если нажать кнопку «Назад» в браузере, то мы вернёмся обратно к началу страницы и адресная строка примет исходный вид. Это позволяет передавать ссылки вида webref.ru/layout/#title, при открытии такой ссылки в браузере будет загружена страница и сделан переход к элементу с идентификатором title.

#### Относительные и абсолютные пути ссылок

Многие, когда начинают только изучать язык гипертекстовой разметки, сталкиваются с трудностями при указании путей к файлам, ведь не всегда страница или даже изображение будет в той же папке, что и текущий HTML документ. Давайте детально разберем как правильно задавать пути, научимся отличать относительные пути от абсолютных и рассмотрим отличия корня веб-сервера от корня вашего диска.

В приведенном в начале статьи примере используется абсолютный адрес пути. Название абсолютный связано с тем, что мы задаем полный веб-адрес страницы в качестве пути к файлу. Ниже приведены примеры указания абсолютных путей:

| <a href = "https://www.yandex.ru">Содержимое элемента</a> <!-- указываем полный путь (протокол https) --> <a href = "http://www.yandex.ru">Содержимое элемента</a> <!-- указываем полный путь (протокол http) --> <a href = "//www.yandex.ru">Содержимое элемента</a> <!-- допускается указывать полный путь без протокола --> |
| --- |

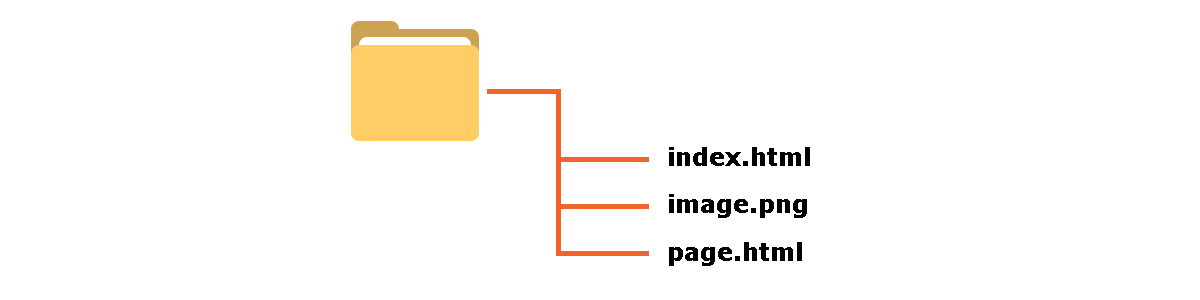
*Обращаю Ваше внимание, что в большинстве случаев, абсолютные пути ссылок применяются для страниц или файлов, которые находятся на других ресурсах (располагаются не на том же сервере, что и страница с которой ссылаются).*

*Если страница или файл на который ссылаются расположен на том же сервере, что и страница с которой ссылаются, то, как правило используются относительные адреса ссылок. Основная особенность относительных ссылок заключается в том, что они размещаются относительно корня сайта или относительно текущего HTML документа.*

#### Путь относительно текущего документа

Предлагаю для начала разобраться с путями относительно текущего HTML документа. Независимо в каких дебрях находится файл, путь вы будете задавать относительно этого файла, этим методом вы будете пользоваться и в практическом задании.

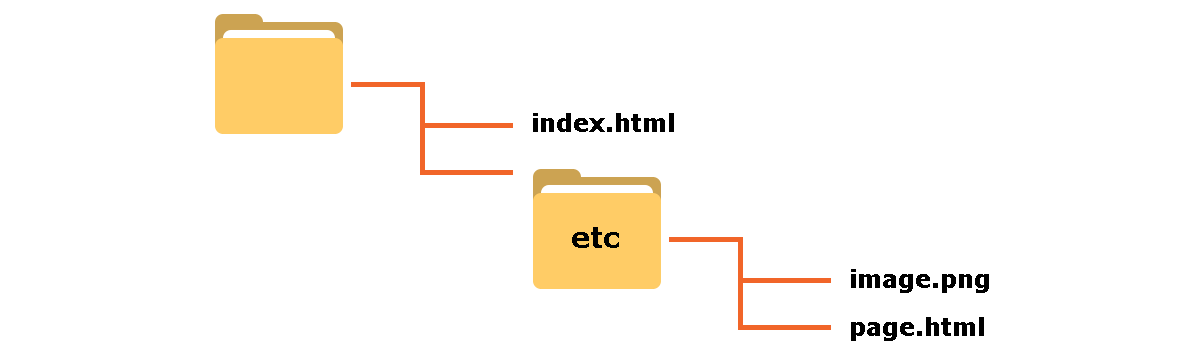
В примерах, когда мы размещали на нашей странице изображения, то мы указывали путь, как раз относительно текущего документа. Давайте рассмотрим аналогичный пример, но также укажем ссылку на html документ, расположенный в той же папке, что и сам файл:



Пример подключения файлов:

| <!DOCTYPE html> <html>  <head>  <title>Пример подключения файлов к HTML документу из той же папки<title>  </head>  <body>  <img src = "image.png" alt = "someimage info"> <!-- размещаем изображение -->  <a href = "page.html">Содержимое элемента</a> <!-- ссылаемся на страницу -->  </body> </html> |
| --- |

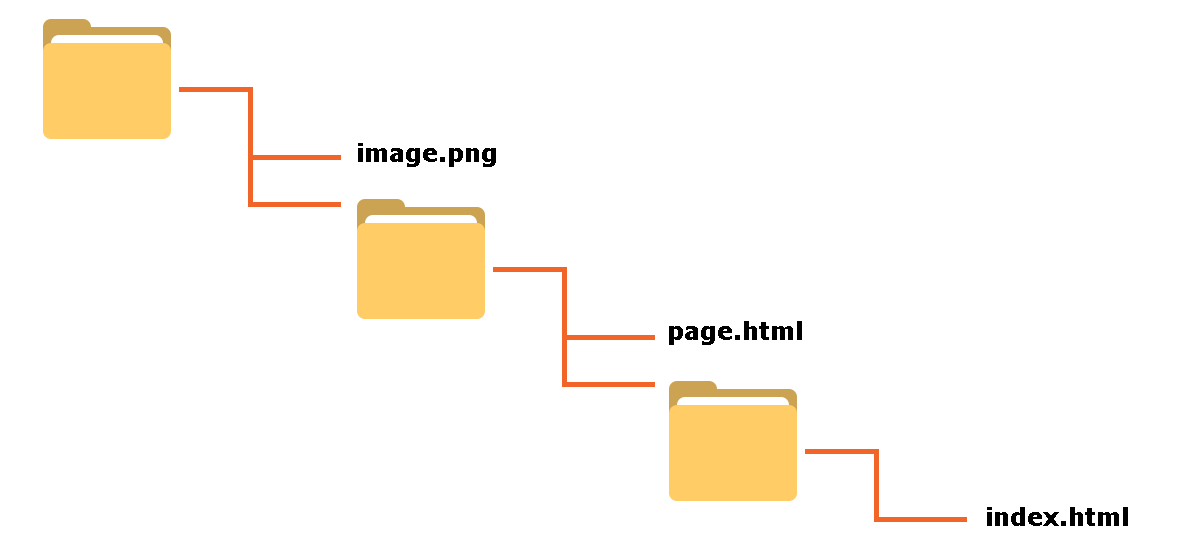
Перейдем к следующему примеру, допустим у нас есть файл index.html, в этом же каталоге есть папка (каталог) с именем etc из которой нам необходимо подключить изображение и сделать ссылку на документ page.html:



| <!DOCTYPE html> <html>  <head>  <title>Пример подключения файлов из дочернего каталога<title>  </head>  <body>  <img src = "etc/image.png" alt = "someimage info"> <!-- размещаем изображение из дочернего каталога -->  <a href = "etc/page.html">Содержимое элемента</a> <!-- ссылаемся на страницу, из дочернего каталога -->  </body> </html> |
| --- |

Обратите внимание, как мы указываем путь к нашим файлам - мы задаем название каталога и через прямой слеш указываем название необходимого нам файла. Данный каталог является по отношению к текущему каталогу дочерним и чтобы подключить необходимые файлы мы спускаемся на один уровень ниже (название\_каталога/необходимый\_файл).

Ну и заключительный пример, в котором нам необходимо подключить к нашей странице изображение, которое находится в каталоге, который расположен на два уровня выше текущей страницы и необходимо добавить ссылку, которая находится на один уровень выше текущей страницы:



| <!DOCTYPE html> <html>  <head>  <title>Пример подключения файлов из родительского каталога<title>  </head>  <body>  <img src = "../../image.png" alt = "someimage info"> <!-- размещаем изображение, которое находится на 2 уровня выше -->  <a href = "../page.html">Содержимое элемента</a> <!-- ссылаемся на страницу, которая находится на 1 уровень выше -->  </body> </html> |
| --- |

Обратите внимание, что если файл находится в родительской папке, то необходимо использовать символы ../ это позволит подняться на один уровень вверх. Вы можете, используя символы ../ подниматься вверх пока не доберетесь до корневой папки Вашего сайта или жесткого диска.

### Семантическая разметка

Теги <div> (рассматриваемый нами ранее) и <span> представляют из себя универсальные пустые контейнеры, которые необходимо заполнить каким-либо содержанием, либо сгруппировать вложенные элементы для их дальнейшей стилизации средствами CSS,, и при необходимости динамически манипулировать ими с использованием Javascript.

Обращаю Ваше внимание на то, что элементы <div> и <span> сами по себе не оказывают никакого влияния на другие элементы страницы.

Тег <div> определяет любой отдельный блок содержимого и предназначен для деления веб-страницы на фрагменты. Вы можете объединить любой набор логически связанных элементов в единственном блоке <div>.

Этот элемент является блочным и это означает, что его содержимое всегда будет начинаться с новой строки, при этом элемент занимает всю доступную ширину. Не так давно, этот элемент занимал абсолютный топ, посвященный делению страницы на логические составляющие, но с приходом HTML 5 было добавлено достаточно семантических элементов, которые позволяют в некоторых ситуациях заменить тег <div> на более осмысленный, но об этом чуть позднее.

В отличии от блочного элемента <div> тег <span> является строчным и применяется к внутренним (inline) элементам страницы, то есть к словам, фразам, которые находятся в пределах абзаца или небольшого фрагмента текста, оглавления и тому подобное.

Тег <span> вы можете использовать для таких задач как:

* Выделение участка текста определённым цветом, фоном или даже фоновым изображением.
* Изменение отдельным словам или фразам размера шрифта, семейства шрифта или типа шрифта.
* Создание различных форматирующих стилей для выбранного участка текста.
* Использование скриптовых языков программирования к выбранным текстовым участкам, например подсветка синтаксиса кода и так далее.

*Мы еще не затрагивали тему CSS, но в случае с тегом span, не упомянуть его было бы не правильно.*

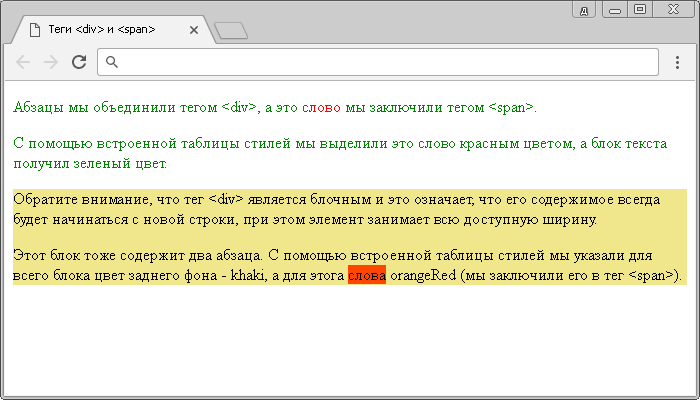
Пример использования:

| <div style="color:green">  <p>Абзацы мы объединили тегом &lt;div&gt;, а это <span style="color:red">слово</span> мы заключили тегом &lt;span&gt;.</p>  <p>С помощью встроенной таблицы стилей мы выделили это слово красным цветом, а блок текста получил зеленый цвет.</p> </div> <div style="background-color:khaki">  <p>Обратите внимание, что тег &lt;div&gt; является блочным и это означает, что его содержимое всегда будет начинаться с новой строки, при этом элемент занимает всю доступную ширину.</p>  <p>Этот блок тоже содержит два абзаца. С помощью встроенной таблицы стилей мы указали для всего блока цвет заднего фона - khaki, а для этого <span style="background-color:orangered">слова</span> orangeRed (мы заключили его в тег &lt;span&gt;).</p> </div> |
| --- |

В этом примере мы:

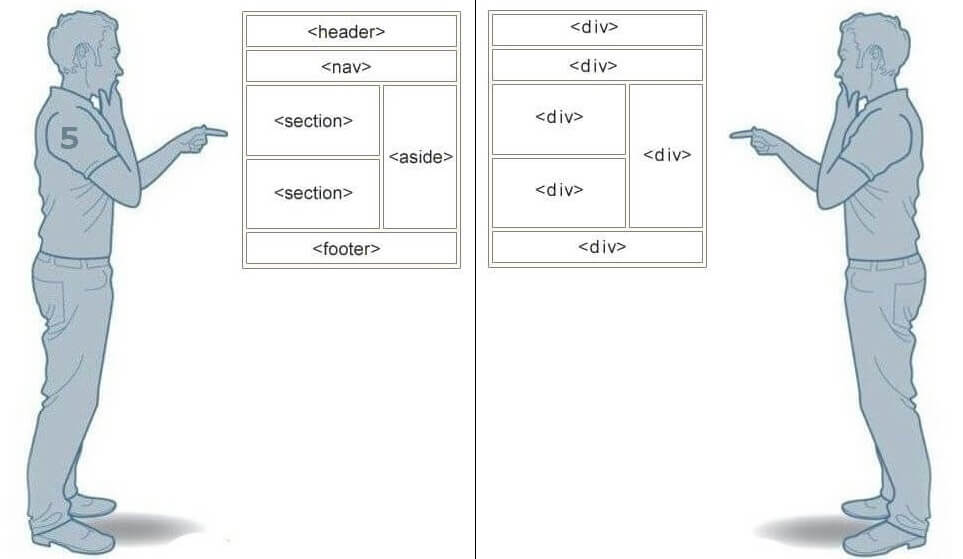
* Блочными элементами <div> создали два блока, содержащие по два абзаца текста внутри (элементы <p>).
* Для первого блока с помощью встроенной таблицы стилей мы указали, что все элементы получат цвет текста зеленый, а одно слово, которое мы поместили внутри тега <span> красный.
* Для второго блока с помощью встроенной таблицы стилей мы указали, что весь блок получит цвет заднего фона хаки, а одно слово, которое мы поместили внутри тега <span> оранжево-красный.

Результат нашего примера:



#### Теги разметки страницы в HTML 5

Тег <div> имеет довольно общий характер — это просто элемент на уровне блока, используемый для разбиения страницы на разделы. Одна из целей HTML 5 заключается в предоставлении в распоряжение разработчиков широкого выбора из других, семантически более осмысленных тегов.



*Человек, который использует только тег <div>.*

HTML 5 включает в себя множество различных тегов, чьи имена отражают тип их содержимого, и эти теги могут использоваться вместо тега <div>. HTML 5 предлагает новые блочные семантические элементы для определения различных частей веб-страницы, давайте их детально рассмотрим.

| <header>"Шапка" статьи или всей страницы</header> <nav>Панель навигации</nav> <main>Основное содержимое</main> <aside>Косвенный контент</aside> <footer>"Подвал" статьи или страницы</footer> <section>Смысловой раздел документа</section> <article>Независимая, отделяемая смысловая часть документа</article> <figure>Иллюстрация к статье</figure> <figcaption>Подпись к иллюстрации</figcaption> |
| --- |

##### Тег <header>

Тег <header> (верхний колонтитул) является признаком верхнего колонтитула или баннера верхней части страницы или осмысленной её части (например статьи), которая обычно содержит заголовок страницы с подзаголовком, авторскую информацию и так далее. Допускается размещать несколько элементов <header> в одном документе.

| <header>  <img src = "logo.png" alt = "logo"> <!-- изображение с логотипом -->  <h1>Заголовок первого уровня</h1> <!-- заголовок первого уровня --> </header> |
| --- |

*Обращаю Ваше внимание, что запрещается помещать элемент <header>html5 внутрь таких элементов, как <footer>html5 (нижний колонтитул), <address>html5 (определяет контактную информацию) или внутрь другого элемента <header>html5.*

##### Тег <nav>

Тег <nav> (навигация) используется для обозначения содержимого в виде основных навигационных ссылок. Документ может иметь несколько элементов <nav>, например, один для навигации по сайту, а второй для навигации по странице.

| <nav> <!-- начало навигации -->  <ul> <!-- маркированный список -->  <li><a href = "#">Домой</a></li> <!-- ссылка в элементе списка -->  <li><a href = "#">HTML</a></li> <!-- ссылка в элементе списка -->  <li><a href = "#">CSS</a></li> <!-- ссылка в элементе списка -->  </ul> <!--конец списка--> </nav> <!-- конец навигации --> |
| --- |

*Обратите внимание, что не все ссылки в документе следует помещать внутрь тега <nav>, элемент предназначен только для крупных навигационных блоков. Не используйте элемент <nav> внутри тега <footer> (нижний колонтитул). Программы чтения с экрана для пользователей с ограниченными возможностями, могут использовать этот элемент, чтобы определить, нужно ли опустить начальную визуализацию этого содержимого.*

##### Тег <aside>

Тег <aside> (отступление) предназначен для обозначения содержимого, относящегося к окружающему этот тег содержимому. Элемент <aside> представляет собой раздел страницы с контентом, который может рассматриваться отдельно от основного содержания. В этих разделах часто размещаются боковые колонки, рекламный контент, биографические данные, веб-приложения, информация о профиле пользователя, пометки на полях в печатном журнале и так далее.

| <article> <!-- начало статьи -->  <p>Сегодня мы с семьей пересматривали фильм "Один дома 2"...</p>  <aside> <!-- отступление (выносим информацию, например, на поля) -->  <p>В одной из сцен фильма снялся известный актер Иван Петров</p>  </aside> </article> <!-- конец статьи --> |
| --- |

##### Тег <article>

Тег <article> (статья) применяется для обозначения раздела страницы, содержащего завершенную, независимую композицию. Данный элемент может содержать в себе такую информацию как запись блога, товар в интернет-магазине, пост на форуме, газетная статья, любой другой независимый элемент содержимого или просто основной текст страницы.

Элемент должен иметь в качестве дочернего (вложенного) элемента заголовок от <h2> до <h6> (если до этого не использовался <h1>, то допускается его разовое размещение). Внутри тега <article> могут содержаться другие одноименные элементы с близким по смыслу содержимым.

| <article> <!-- начало основной статьи -->  <h1>Статьи о птицах</h1><!-- заголовок первого уровня (если был использован в документе, то h2, а во вложенных h3) -->  <p>Вводная информация о птицах...</p>  <article> <!-- начало первой вложенной статьи -->  <h2>Статья о воробье</h2>  <p>Информация про воробьев...</p>  </article> <!-- конец первой вложенной статьи -->  <article> <!-- начало второй вложенной статьи -->  <h2>Статья о синице</h2>  <p>Информация про синиц...</p>  </article><!-- конец второй вложенной статьи --> </article> <!-- конец основной статьи --> |
| --- |

##### Тег <section>

Тег <section>html5 (раздел) служит для группировки взаимосвязанного содержимого.

Не используйте элемент <section> в качестве универсального контейнера, для этих целей подходит элемент <div>. Раздел должен логически отображать структуру документа, например, вы можете разбить содержимое главной страницы на три раздела: вводную информацию о сайте, контактную информацию и самые последние новости.

*Данный тег часто используется при верстке целевых страниц (англ. landing page, "посадочная страница") для логического разделения страницы на разделы.*

*Элемент должен иметь в качестве дочернего (вложенного) элемента заголовок от <h2> до <h6> (если до этого не использовался <h1>, то допускается его разовое размещение).*

| <section> <!-- начало первой вложенной статьи -->  <h2>Первый раздел</h2>  <p>Информация внутри раздела</p> </section> <!-- конец первой вложенной статьи --> <section> <!-- начало второй вложенной статьи -->  <h2>Второй раздел</h2>  <p>Информация внутри раздела</p> </section><!-- конец второй вложенной статьи --> |
| --- |

##### Совместное использование тегов <section> и <article>

Обратите внимание, что допускается в качестве дочернего (вложенного) элемента тега <article> использовать тег <section> (раздел), как бы разбивая по смыслу содержимое. При этом необходимо, чтобы элемент <article>html5 и вложенные элементы <section> имели в качестве дочернего (вложенного) элемента заголовок от <h2> до <h6> (если до этого не использовался <h1>, то допускается его разовое размещение внутри <article>).

По аналогии с вышерассмотренным примером допускается помещать элементы <article> внутрь элементов <section>, формируя при этом разделы с тематической информацией. При этом на элементы действуют те же правила по размещению заголовков:

Допускается помещать элементы <section> (раздел) в другие элементы <section>, но при этом рекомендуется учитывать следующую структуру документа:

| <body>  <h1>Заголовок первого уровня</h1>  <section>  <h2>Заголовок второго уровня</h2>  <section>  <h3>Заголовок третьего уровня</h3>  </section>  </section> |
| --- |

##### Тег <footer>

Тег <footer> (нижний колонтитул) содержит информацию, которая обычно помещается в нижнем колонтитуле страницы, например сведения об авторских правах, другую правовую информацию, некоторые ссылки для навигации по сайту и тому подобное.

Обращаю Ваше внимание, что на количество тегов <footer> на одной странице ограничений не накладывается, вы можете, допустим, поместить нижний колонтитул внутри тега <article>, чтобы хранить в нем информацию, относящуюся к статье, например сноски, ссылки или выписки.

| <body>  <h1>Важный заголовок</h1>  <article> <!-- начало первой статьи -->  <h2>Статья о бытие</h2>  <p>Информация про бытие...</p>  <footer>Сноски, ссылки и тому подобное <!-- "подвал" статьи -->  <address>Информация об авторе статьи</address>  </footer>  </article><!-- конец первой статьи -->  <article> <!-- начало второй статьи -->  <h2>Статья о бытие 2</h2>  <p>Информация про бытие 2...</p>  <footer>Сноски, ссылки и тому подобное <!-- "подвал" статьи -->  <address>Информация об авторе статьи</address>  </footer>  </article> <!-- конец второй статьи -->  <footer>сведения об авторских правах, навигация по сайту, счетчики... <!-- "подвал" сайта -->  <address>Контактные данные автора сайта</address>  </footer> </body> |
| --- |

##### 

##### Тег <main>

Тег <main> предназначен для основного содержимого документа (основной контент). Контент внутри элемента должен быть уникальным для всего документа и не должен содержать элементы, которые повторяются в различных документах (боковые панели, навигационные ссылки, информация об авторских правах, логотип сайта, формы поиска и тому подобное). Допускается использование элемента только один раз в одном документе.

*Обращаю Ваше внимание, что тег <main> не должен быть потомком таких блоков как (не должен быть вложен в них):*

* *Тег <nav> (навигация).*
* *Тег <header> (верхний колонтитул).*
* *Тег <footer> (нижний колонтитул).*
* *Тег <aside> (отступление).*
* *Тег <article> (статья).*

#### Разметка иллюстраций в HTML 5

Тег <figure> применяется для обозначения контента, иллюстрирующего или поддерживающего определенную идею текста. Тег <figure> может содержать изображение, видеоролик, схему, фрагмент кода, диаграмму или даже таблицу — почти все, что может встретиться в потоке веб-контента и должно восприниматься как автономная единица.

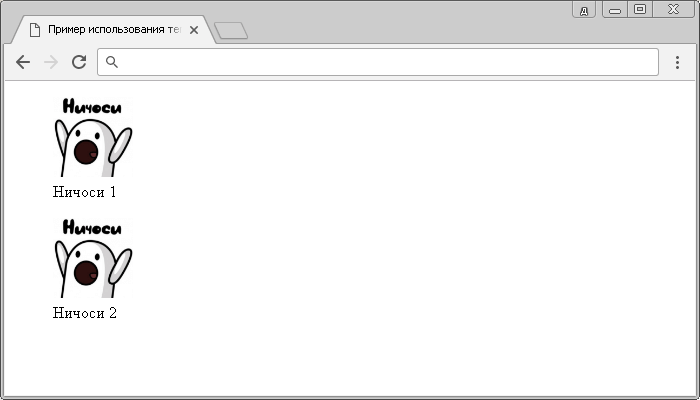
Благодаря тегу <figcaption> вы можете вывести пояснение к содержимому, которое находится внутри тега <figure>. Тег <figcaption> должен размещаться как первый, или последний элемент внутри тега <figure>.

Давайте рассмотрим пример использования:

| <!DOCTYPE html> <html>  <head>  <title>Пример использования тега <figure></title>  </head>  <body>  <main>  <figure>  <img src = "10.jpg" alt = "ничоси" width = "80" height = "80">  <figcaption>Ничоси 1</figcaption>  </figure>  <figure>  <img src = "10.jpg" alt = "ничоси" width = "80" height = "80">  <figcaption>Ничоси 2</figcaption>  </figure>  </main>  </body> </html> |
| --- |

В этом примере основное содержимое документа поместили внутри тега <main>, в который мы поместили два элемента <figure>, каждый из которых содержит внутри себя изображение (тег <img>) шириной и высотой 80 пикселей. Кроме того тегом <figcaption> мы вывели текстовое пояснение к содержимому каждого элемента <figure>. Элемент <figcaption> мы разместили перед закрывающим тегом </figure>.

Результат нашего примера:



*Обратите внимание, что если вы разместите элемент <figcaption> первым вложенным элементом тега <figure>, то пояснение будет отображаться сверху изображения.*

### Самостоятельная работа

Попробуйте применить полученные знания на практике. Создайте html-файл с заголовком "Задача 5", описывающий фильм, мультфильм, персонажа или другую личность. Постарайтесь максимально использовать полученные знания, добавьте изображение, описание и другие характеристики. Используйте семантическую разметку, списки и все другое, что потребуется для реализации задачи.

# ШПАРГАЛКА

## HTML

| **Элемент** | **Описание** |
| --- | --- |
| <!--...--> | Используется для добавления комментариев. |
| <!DOCTYPE> | Объявляет тип документа и предоставляет основную информацию для браузера — его язык и версия. |
| <a> | Создаёт гипертекстовые ссылки. |
| <abbr> | Определяет текст как аббревиатуру или акроним. Поясняющий текст задаётся с помощью атрибута title. |
| <address> | Задает контактные данные автора/владельца документа или статьи. Отображается в браузере курсивом. |
| <area> | Представляет собой гиперссылку с текстом, соответствующей определенной области на карте-изображении или активную область внутри карты-изображения. Всегда вложен внутрь элемента <map>. |
| <article> | Раздел контента, который образует независимую часть документа или сайта, например, статья в журнале, запись в блоге, комментарий. |
| <aside> | Представляет контент страницы, который имеет косвенное отношение к основному контенту страницы/сайта. |
| <audio> | Загружает звуковой контент на веб-страницу. |
| <b> | Задает полужирное начертание отрывка текста, не придавая акцент или важность выделенному. |
| <base> | Задает базовый адрес (URL), относительно которого вычисляются все относительные адреса. Это поможет избежать проблем при переносе страницы в другое место, так как все ссылки будут работать, как и прежде. |
| <bdi> | Изолирует отрывок текста, написанный на языке, в котором чтение текста происходит справа налево, от остального текста. |
| <bdo> | Отображает текст в направлении, указанном в атрибуте dir, переопределяя текущее направление написания текста. |
| <blockquote> | Выделяет текст как цитату, применяется для описания больших цитат. |
| <body> | Представляет тело документа (содержимое, не относящееся к метаданным документа). |
| <br> | Перенос текста на новую строку. |
| <button> | Создает интерактивную кнопку. Элемент может содержать текст или изображение. |
| <canvas> | Холст-контейнер для динамического отображения изображений, таких как простые изображения, диаграммы, графики и т.п. Для рисования используется скриптовый язык JavaScript. |
| <caption> | Добавляет подпись к таблице. Вставляется сразу после открывающего тега <table>. |
| <cite> | Используется для указания источника цитирования. Отображается курсивом. |
| <code> | Представляет фрагмент программного кода, отображается шрифтом семейства monospace. |
| <col> | Выбирает для форматирования один или несколько столбцов таблицы, не содержащих информацию одного типа. |
| <colgroup> | Создает структурную группу столбцов, выделяющую множество логически однородных ячеек. |
| <data> | Элемент используется для связывания значения атрибута value, которое представлено в машиночитаемом формате и может быть обработано компьютером, с содержимым элемента. |
| <datalist> | Элемент-контейнер для выпадающего списка элемента <input>. Варианты значений помещаются в элементы <option>. |
| <dd> | Используется для описания термина из элемента <dt>. |
| <del> | Помечает текст как удаленный, перечёркивая его. |
| <details> | Создаёт интерактивный виджет, который пользователь может открыть или закрыть. Представляет собой контейнер для контента, видимый заголовок виджета помещается в элемент <summary>. |
| <dfn> | Определяет слово как термин, выделяя его курсивом. Текст, идущий следом, должен содержать расшифровку этого термина. |
| <dialog> | Интерактивный элемент, с которым взаимодействует пользователь для выполнения задачи, например, диалоговое окно, инспектор или окно. Без атрибута open не виден для пользователя. |
| <div> | Элемент-контейнер для разделов HTML-документа. Используется для группировки блочных элементов с целью форматирования стилями. |
| <dl> | Элемент-контейнер, внутри которого находятся термин и его описание. |
| <dt> | Используется для задания термина. |
| <em> | Выделяет важные фрагменты текста, отображая их курсивом. |
| <embed> | Элемент-контейнер для встраивания внешнего интерактивного контента или плагина. |
| <fieldset> | Группирует связанные элементы в форме, рисуя рамку вокруг них. |
| <figcaption> | Заголовок/подпись для элемента <figure>. |
| <figure> | Самодостаточный элемент-контейнер для такого контента как иллюстрации, диаграммы, фотографии, примеры кода, обычно с подписью. |
| <footer> | Определяет завершающую область (нижний колонтитул) документа или раздела. |
| <form> | Форма для сбора и отправки на сервер информации от пользователей. Не работает без атрибута action. |
| <h1-h6> | Создают заголовки шести уровней для связанных с ними разделов. |
| <head> | Элемент-контейнер для метаданных HTML-документа, таких как <title>, <meta>, <script>, <link>, <style>. |
| <header> | Секция для вводной информации сайта или группы навигационных ссылок. Может содержать один или несколько заголовков, логотип, информацию об авторе. |
| <hr> | Горизонтальная линия для тематического разделения параграфов. |
| <html> | Корневой элемент HTML-документа. Сообщает браузеру, что это HTML-документ. Является контейнером для всех остальных html-элементов. |
| <i> | Выделяет отрывок текста курсивом, не придавая ему дополнительный акцент. |
| <iframe> | Создает встроенный фрейм, загружая в текущий HTML-документ другой документ. |
| <img> | Встраивает изображения в HTML-документ с помощью атрибута src, значением которого является адрес встраиваемого изображения. |
| <input> | Создает многофункциональные поля формы, в которые пользователь может вводить данные. |
| <ins> | Выделяет текст подчеркиванием. Применяется для выделения изменений, вносимых в документ. |
| <kbd> | Выделяет текст, который должен быть введён пользователем с клавиатуры, шрифтом семейства monospace. |
| <label> | Добавляет текстовую метку для элемента <input>. |
| <legend> | Заголовок элементов формы, сгруппированных с помощью элемента <fieldset>. |
| <li> | Элемент маркированного или нумерованного списка. |
| <link> | Определяет отношения между документом и внешним ресурсом. Также используется для подключения внешних таблиц стилей. |
| <main> | Контейнер для основного уникального содержимого документа. На одной странице должно быть не более одного элемента <main>. |
| <map> | Создаёт активные области на карте-изображении. Является контейнером для элементов <area>. |
| <mark> | Выделяет фрагменты текста, помечая их желтым фоном. |
| <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. |
| <meter> | Индикатор измерения в заданном диапазоне. |
| <nav> | Раздел документа, содержащий навигационные ссылки по сайту. |
| <noscript> | Определяет секцию, не поддерживающую сценарий (скрипт). |
| <object> | Контейнер для встраивания мультимедиа (например, аудио, видео, Java-апплеты, ActiveX, PDF и Flash). Также можно вставить другую веб-страницу в текущий HTML-документ. Для передачи параметров встраиваемого плагина используется элемент <param>. |
| <ol> | Упорядоченный нумерованный список. Нумерация может быть числовая или алфавитная. |
| <optgroup> | Контейнер с заголовком для группы элементов <option>. |
| <option> | Определяет вариант/опцию для выбора в раскрывающемся списке <select>, <optgroup> или <datalist>. |
| <output> | Поле для вывода результата вычисления, рассчитанного с помощью скрипта. |
| <p> | Параграфы в тексте. |
| <param> | Определяет параметры для плагинов, встраиваемых с помощью элемента <object>. |
| <picture> | Элемент-контейнер, содержащий один элемент <img> и ноль или несколько элементов <source>. Сам по себе ничего не отображает. Дает возможность браузеру выбирать наиболее подходящее изображение. |
| <pre> | Выводит текст без форматирования, с сохранением пробелов и переносов текста. Может быть использован для отображения компьютерного кода, сообщения электронной почты и т.д. |
| <progress> | Индикатор выполнения задачи любого рода. |
| <q> | Определяет краткую цитату. |
| <ruby> | Контейнер для Восточно-Азиатских символов и их расшифровки. |
| <rb> | Определяет вложенный в него текст как базовый компонент аннотации. |
| <rt> | Добавляет краткую характеристику сверху или снизу от символов, заключенных в элементе <ruby>, выводится уменьшенным шрифтом. |
| <rtc> | Отмечает вложенный в него текст как дополнительную аннотацию. |
| <rp> | Выводит альтернативный текст в случае если браузер не поддерживает элемент <ruby>. |
| <s> | Отображает текст, не являющийся актуальным, перечеркнутым. |
| <samp> | Используется для вывода текста, представляющего результат выполнения программного кода или скрипта, а также системные сообщения. Отображается моноширинным шрифтом. |
| <script> | Используется для определения сценария на стороне клиента (обычно JavaScript). Содержит либо текст скрипта, либо указывает на внешний файл сценария с помощью атрибута src. |
| <section> | Определяет логическую область (раздел) страницы, обычно с заголовком. |
| <select> | Элемент управления, позволяющий выбирать значения из предложенного множества. Варианты значений помещаются в <option>. |
| <small> | Отображает текст шрифтом меньшего размера. |
| <source> | Указывает местоположение и тип альтернативных медиаресурсов для элементов <picture>, <video>, <audio>. |
| <span> | Контейнер для строчных элементов. Можно использовать для форматирования отрывков текста, например, выделения цветом отдельных слов. |
| <strong> | Расставляет акценты в тексте, выделяя полужирным. |
| <style> | Подключает встраиваемые таблицы стилей. |
| <sub> | Задает подстрочное написание символов, например, индекса элемента в химических формулах. |
| <summary> | Создаёт видимый заголовок для элемента <details>. Отображается с закрашенным треугольником, кликнув по которому можно просмотреть подробности заголовка. |
| <sup> | Задает надстрочное написание символов. |
| <table> | Элемент для создания таблицы. |
| <tbody> | Определяет тело таблицы. |
| <td> | Создает ячейку таблицы. |
| <template> | Используется для объявления фрагментов HTML-кода, которые могут быть клонированы и вставлены в документ скриптом. Содержимое элемента не является его дочерним элементом. |
| <textarea> | Создает большие поля для ввода текста. |
| <tfoot> | Определяет нижний колонтитул таблицы. |
| <th> | Создает заголовок ячейки таблицы. |
| <thead> | Определяет заголовок таблицы. |
| <time> | Определяет дату/время. |
| <title> | Заголовок HTML-документа, отображаемый в верхней части строки заголовка браузера. Также может отображаться в результатах поиска, поэтому это следует принимать во внимание предоставление названия. |
| <tr> | Создает строку таблицы. |
| <track> | Добавляет субтитры для элементов <audio> и <video>. |
| <u> | Выделяет отрывок текста подчёркиванием, без дополнительного акцента. |
| <ul> | Создает маркированный список. |
| <var> | Выделяет переменные из программ, отображая их курсивом. |
| <video> | Добавляет на страницу видео-файлы. Поддерживает 3 видео формата: MP4, WebM, Ogg. |
| <wbr> | Указывает браузеру возможное место разрыва длинной строки. |

### 

### 

## CSS

### Текст

| text-align: center; | Выравнивание текста по центру; |
| --- | --- |
| text-align: left; | Выравнивание текста по левому краю; |
| text-align:right; | Выравнивание текста по правому краю; |
| text-decoration: none; | Текст без подчёркивания; |
| text-decoration: underline; | Подчёркнутый текст; |
| text-decoration: overline; | Черта над текстом; |
| text-decoration: line-through; | Зачёркнутый текст; |
| text-decoration: blink; | Мигающий текст; |
| text-transform: uppercase; | Делает все буквы заглавными; |
| text-transform: capitalize; | Делает все первые буквы заглавными; |
| text-indent: 30px; | Красная строка; |
| word-spacing: 30px; | Задаёт расстояние между словами; |
| letter-spacing:2px; | Задаёт расстояние между буквами; |
| line-height: 10px; | Задаёт фиксированную высоту строки; |
| direction: rtl; | Задаёт направление; текста справа налево. |
| color: #00ff00; | Цвет текста; |
| text-schadow: | Тень текста; |

### Шрифт

| font-style: italic; | Шрифт курсив; |
| --- | --- |
| font-weight: bold; | Жирный шрифт; |
| font-weight: lighter; | Тонкий шрифт; |
| font-size: 16px; | Размер шрифта; |
| font-family: Georgia | Название шрифта; |
| font-variant: small-caps; | Делает все буквы прописными; |
| font:italic bold 16px Georgia; | Сокращение записи для свойства font |

### Внешние и внутренние отступы

| margin-top: 20px; | Внешний отступ сверху; |
| --- | --- |
| margin-right: 20px; | Внешний отступ справа; |
| margin-bottom: 20px; | Внешний отступ снизу; |
| margin-left: 20px; | Внешний отступ слева; |
| margin: 20px 20px 20px 20px; | Сокращённая запись для внешних отступов, первая цифра отступ сверху, и далее по часовой стрелке; |
| padding-top: 10px; | Внутренний отступ сверху; |
| padding-right:10px; | Внутренний отступ справа; |
| padding-bottom: 10px; | Внутренний отступ снизу; |
| padding-left: 10px; | Внутренний отступ слева; |
| padding: 10px 10px 10px 10px; | Сокращение записи для свойства padding, расположение аналогично margin |

### Свойства фона

| background-color: #00ff00; | Цвет фона элемента; |
| --- | --- |
| background-position: center; | Размещает фоновое изображение по центру; |
| background-position: 100px 150px; | Размещает фоновое изображение, первая цифра — отступ сверху (top), вторая — отступ слева left; |
| background-repeat:repeat; | Изображение повторяется пока не займёт всё пространство блока; |
| background-repeat: no-repeat | Изображение не повторяется; |
| background-repeat:repeat-y | Изображение повторяется по вертикали; |
| background-repeat:repeat-x; | Изображение повторяется по горизонтали; |
| background-attachment: fixed; | Изображение фиксируется и остаётся на месте при прокрутке страницы; |
| background-image:url(images/im.png) | Вставляет изображение в элемент; |
| background: #00ff00 url(images/im.png) no-repeat fixed center; | Сокращение записи для background; |
| background-size: | Масштабирует фоновое изображение; |

### Ширина и высота

| width: 100px; | Ширина элемента; |
| --- | --- |
| min-width: 100px; | Минимальная ширина элемента; |
| max-width: 100px; | Максимальная ширина элемента; |
| height: 100px; | Высота элемента; |
| min-height: 100px; | Минимальная высота элемента; |
| max-height: 100px; | Максимальная высота элемента; |

### Рамки

| border: 3px; | Рамка элемента; |
| --- | --- |
| border-color:#ff0000; | Цвет рамки; |
| border-style: solid; | Стиль рамки; |
| border: 5px solid #ff0000; | Сокращение записи для рамки; |
| outline: 5px solid #00FF00; | Задаёт рамку за внешней границей элемента; |
| border-radius: #00FF00 25px; | Создаёт рамку с закруглёнными углами вокруг элемента; |

### Свойства списка

| list-style-type: | Определяет вид маркера в списке. Применяемые значения:  none — без маркера;  circle — круг;  disc — заполненный круг;  square — квадрат;  upper-alpha — A B C D …;  upper-roman — I II III IV V …;  lower-alpha — a b c d …;  decimal-leading-zero — 01 02 03 04 …;  none — отменяет маркер |
| --- | --- |
| list-style-image: url(images/im.png) | Устанавливает маркер-изображение; |
| list-style-position: inside; | Размещает маркер-изображение внутри блока списка; |
| list-style:square inside url(images/im.png); | Сокращение записи для list-style, square указывается на случай недоступности картинки; |

### Позиционирование

| position: absolute; | Размещает элемент относительно границ экрана монитора. Применяется со свойствами: top:, left:, right:, bottom:, в которых указываются координаты позиционирования |
| --- | --- |
| position: relative; | Применяется к элементу, относительно которого позиционируется другой элемент; |
| position: fixed; | Фиксирует элемент в указанной части экрана. Элемент не реагирует на прокрутку |
| x-index: ; | Накладывает элементы друг на друга по оси х; |
| y-index: ; | Накладывает элементы друг на друга по оси у; |
| z-index: ; | Накладывает элементы друг на друга по оси z; |
| float: right; | Задаёт обтекание слева; |
| float: left; | Задаёт обтекание справа; |
| clear: both; | Запрещает обтекание; |
| clip: rect(10px,50px); | Обрезает изображение квадратом; |

### Видимость

| visibility: hidden; | Делает элемент невидимым, но сохраняет занимаемое им место; |
| --- | --- |
| display: none; | Временно удаляет элемент со страницы, и другие элементы могут занять его место; |
| display: block; | Делает элемент блочным; |
| display: inline; | Делает элемент встроенным; |
| display: inline-block; | Делает элемент встроенным, но с ддоступными свойствами блочного; |
| overflow: auto; | При необходимости добавляет полосу прокрутки в блок с контентом; |
| opacity: 0; | Делает элемент полностью прозрачным. Прозрачность уменьшается до значения 1; |

### Трансформация и анимация

| transform | поворачивает, сдвигает, наклоняет и масштабирует элемент. |
| --- | --- |
| transform: scale(); | Масштабирует элемент по горизонтали и вертикали. Число в скобках больше единицы увеличивает; элемент, а меньше единицы — уменьшает |
| transform: scaleX(); | Масштабирует элемент по горизонтали; |
| transform: scaleY(); | Ьасштабирует элемент по вертикали |
| transform: skewX(); | Наклоняет элемент на заданный угол в скобках по вертикали влево, отрицательное значение — вправо |
| transform: skewY(); | Наклоняет элемент на заданный угол по горизонтали; |
| transform: translateX(); | Сдвигает элемент по горизонтали |
| transform: translateY(); | Сдвигает элемент по вертикали |
| transform: rotate(); | Поворот элемента вокруг центра на заданный угол |
| transform: rotateY(360deg); | Поворот элемента вокруг вертикальной оси. Применяется в 3D |
| transform: rotateX(360deg); | Поворот элемента вокруг горизонтальной оси. Применяется в 3D |
| transform: matrix() | Создаёт эффект отражения элемента |
| transform-style: preserve-3d | Создаёт 3D эффект |
| transform-origin: x y z; | Задаёт координаты точки относительно центра, вокруг которой будет вращаться элемент |
| transition-property: | Определяет свойство, которое будет анимироваться |
| transition-duration: | Продолжительность эффекта перехода |
| transition-timing-function: | Скорость эффекта перехода |
| transition-delay: | Задержка начала анимации |
| transition: left 3s ease; | Сокращение записи для transition |
| animation | Pадаёт параметры действия для анимации — длительность анимации, повторение, направление, тип движения, шаги. Работает со свойством @keyframes |
| animation-name | Имя анимации |
| animation-duration | Длительность анимации |
| animation-timing-function | Определяет вид анимации |
| animation-iteration-count | Задаёт число повторений анимации |
| Animation-direction | Задаёт направление движения |
| animation-delay | Задаёт задержку анимации перед началом воспроизведения |
| animation: nev 5s infinite alternate; | Сокращение записи для animation |
| @keyframes | Правило, записывается как селектор, внутрь которого помещаются селекторы последовательно определяющие состояние элемента в определённый момент |

### Добавление элементов без html

| content: | Выводит на страницу элементы, которых нет в html коде. Применяется с псевдоклассами ::after и ::before |
| --- | --- |
| content:"" | В кавычки можно добавить текст и он будет выведен на страницу |
| content:attr(); | В скобки обычно добавляется спецсимвол html, изображение которого и выводится в начале или в конце элемента |
| content: url(images/1.png); | Выводит картинку в начале или в конце элемента. Обычно маркер списка. |

### Колончатая вёрстка страницы

| columns: | Разделяет поле страницы на колонки |
| --- | --- |
| column-count: 3; | Количество колонок |
| column-width: 200px; | Ширина колонок |
| column-gap: 1-px; | Ширина пробела между колонками |
| column-span: all; | Указывает количество колонок, которые пересечёт элемент. Применяется обычно для заголовка. |
| column-rule-style: solid; | Создаёт вертикальную разделительную линию между колонками. |
| column-rule-width: 2px; | Ширина разделительной линии |
| column-rule-color: #999; | Цвет разделительной линии |
| column-rule: 2px solid #999; | Краткая запись разделительной линии |

### Медиа запросы

| @media all{Таблица стилей} | Селекторы и свойства css заносятся в фигурные скобки и применяется ко всем типам носителей |
| --- | --- |
| @media screen and (){} | Экран монитора. В круглых скобках указываются дополнительные требования, например размер экрана |
| @media tv and (){} | Телевизор |
| @media print and (){} | Принтеры |
| @media handheld and (){} | Смартфоны |
| @media screen, handheld and (){} | Монитор и смартфон |
| @media screen and (max-width: 600px){} | Для моноторов с разрешением менее 600px |

### Другие @ правила

| @charset «кодировка»; | Устанавливает кодировку для css файла |
| --- | --- |
| @import «имя файла» [типы носителей]; | Импортирует содержимое CSS-файла в текущую стилевую таблицу |

### Цветовые и визуальные фильтры

| filter: blur(3px); | Размытие |
| --- | --- |
| filter: brightness(70%); | Яркость |
| filter: contrast(50%); | Контрастность |
| filter: drop-shadow(2px 3px 5px blue); | Добавляет тень ко всем элементам |
| filter: grayscale(50%); | Обесцвечивание |
| filter: hue-rotate(70deg); | Изменяет цвета в зависимости от заданного угла в цветовом круге |
| filter: invert(100%); | Негатив изображения |
| filter: opacity(30%); | Прозрачность |
| filter: saturate(300%); | Насыщенность цвета. Меньше 100% — уменьшает насыщенность, больше 100% — увеличивает |
| filter: sepia(120%); | Ретро |

### Flexbox

| .flex-container{} | Преобразует любой элемент во флекс-контейнер, а все дочерние элементы во флекс-элементы. Обязательное свойство display: flex; |
| --- | --- |
| justify-content: flex-start; | Дочерние элементы позиционируются от начала контейнера |
| justify-content: flex-end; | Дочерние элементы позиционируются от конца контейнера (справа) |
| justify-content: center; | Дочерние элементы позиционируются по центру контейнера |
| justify-content: space-between; | Элементы выравниваются по главной оси. Первый элемент располагается в начале контейнера, последний в конце, а все остальные равномерно располагаются на свободном пространстве между ними |
| justify-content: space-around; | Свободное пространство делится поровну между элементами, добавляя им отступы |
| justify-content: inherit; | Наследует родительское свойство |
| align-items: stretch; | Флекс элементы занимают всё пространство по высоте. |
| align-items: flex-start; | Все дочерние элементы выравниваются по левому краю и верхней границе flex-бокса |
| align-items: flex-end; | Элементы выравниваются по нижнему краю, относительно нижней границы flex-бокса |
| align-items: center; | Выравнивание по центру |
| align-items: baseline; | Выравнивание по базовой линии |
| align-items: inherit; | Наследует значение родителя |
| flex-direction: row; | Направление главной оси слева направо |
| flex-direction: row-reverse; | Направление главной оси справа налево |
| flex-direction: column; | Направление главной оси сверху вниз |
| flex-direction: column-reverse; | Направление снизу вверх |
| flex-wrap: nowrap; | Flex-элементы не переносятся и располагаются в одну строку слева направо |
| flex-wrap: wrap; | Элементы переносятся и располагаются в несколько рядов слева на право |
| flex-wrap: wrap-reverse; | Элементы переносятся и располагаются в обратном порядке. Перенос осуществляется снизу вверх |
| align-content: stretch; | Строки равномерно растягиваются, заполняя всё доступное пространство |
| align-content: flex-start; | Строки выравниваются по левому краю,относительно верхней границы |
| align-content: flex-end; | Строки выравниваются по левому краю, относительно нижней границы |
| align-content: center; | Выравнивание по центру |
| align-content: space-between; | Строки выравниваются по высоте, свободное пространство распределяется между ними |
| align-content: inherit; | Наследует значение родителя |
| order: -1 или 1; | Ставит любой flex-элемент в начало строки или в конец строки |
| flex-basis: 200px; | Ширина элемента задаётся во всех доступных в веб размерах |
| flex-basis: auto; | Элемент получает базовую ширину, соответствующую ширине контента |
| flex-grow: 1.0 ….; | Устанавливает коэффициент увеличения элемента |
| flex-shrink: 1.0 …; | Устанавливает коэффициент уменьшения элемента |
| align-self: auto; | Выравнивание отдельного элемента. По умолчанию задаётся выравнивание родителя |
| align-self: flex-start; | Отдельный элемент выравнивается по левому краю, относительно верхней границы |
| align-self: flex-end; | Выравнивается по левому краю, относительно нижней границы |
| align-self: center; | Выравнивается по центру |
| align-self: baseline; | Выравнивается по базовой линии слева |
| align-self: stretch; | Элемент растягивается на всю высоту flex-контейнера |