



Разработка веб-страниц на языке разметки HTML с использованием каскадных таблиц стилей CSS

Зуев Александр

Модуль 3

Графика в веб-дизайне. Гиперссылки.

Принципы навигации веб-сайта

Форматы графических файлов в Web

bmp, tif, gif, jpg, png

ВМР (англ. *bitmap*) — формат хранения растровых изображений. Вообще-то не один грамотный веб-мастер не станет использовать графику в этом формате для оформления своего сайта (равно как и для контентного наполнения) - размеры несжатой битовой карты чрезвычайно объемны. Просто забудьте о том, что браузеры поддерживают bmp

TIFF (англ. Tagged Image File Format) — формат хранения растровых графических изображений, который стал популярным форматом для хранения изображений с большой глубиной цвета. Этот формат поддерживается всеми браузерами, но из-за весьма крупных размеров в веб-дизайне не используется. Однако, "тиффы" иногда можно встретить в контентном наполнении различных сайтов - например, в качестве демонстрации "на печать" работ различных фотохудожников.

GIF (англ. Graphics Interchange Format — формат для обмена изображениями). Максимально допустимая глубина цвета - 8 бит (256 цветов). Этот формат очень хорошо подходит для того, что бы отображать на веб-страницах растровую графику. Огромный выбор вариантов компрессии, а так же палитр сделало этот формат одним из самых распространенных в вебе. Ну а возможность делать анимированные картинки (и использовать их в качестве банеров, меню и проч.). Формат совместим на 100% со всеми известными на сегодняшний день операционными системами.

JPEG (Joint Photographic Experts Group) — популярный формат графических файлов, широко используемый при создании сайтов и для хранения изображений. JPEG поддерживает 24-битовый цвет и сохраняет яркость и оттенки цветов в фотографиях неизменными. Данный формат называют сжатием с потерями, поскольку алгоритм JPEG выборочно отвергает данные. Метод сжатия может внести искажения в рисунок, особенно если он содержащий текст, мелкие детали или четкие края. Формат JPEG не поддерживает прозрачность. Когда вы сохраняете фотографию в этом формате, прозрачные пиксели заполняются определенным цветом.

PNG-8 (Portable Network Graphics) — формат по своему действию аналогичен GIF. Использует улучшенный формат сжатия данных. Используется в областях: текст, логотипы, иллюстрации с четкими краями, изображения с градиентной прозрачностью.

PNG-24 — формат, аналогичный PNG-8, но использующий 24-битную палитру цвета. Подобно формату JPEG, сохраняет яркость и оттенки цветов в фотографиях. Подобно форматам GIF и PNG-8 сохраняет детали изображения, как, например, в линейных рисунках, логотипах, или иллюстрациях. Применение: фотографии, рисунки, содержащие прозрачные участки, рисунки с большим количеством цветов и четкими краями изображений.

Тег и его атрибуты (src,
alt, width, height, border)

Элемент `` 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>`.

Тег `` имеет обязательный атрибут `src`, значением которого является абсолютный или относительный путь к изображению:

```

```

Абсолютный путь

src="

https://etikket.ru/images/povsednevnyii-etiket/etiket-na-ulitse1.jpg "

Когда ссылка представляет из себя полный URL файла или страницы, это и есть абсолютный путь

В основном абсолютный путь используется, тогда, когда нужно сослаться на другой сайт.

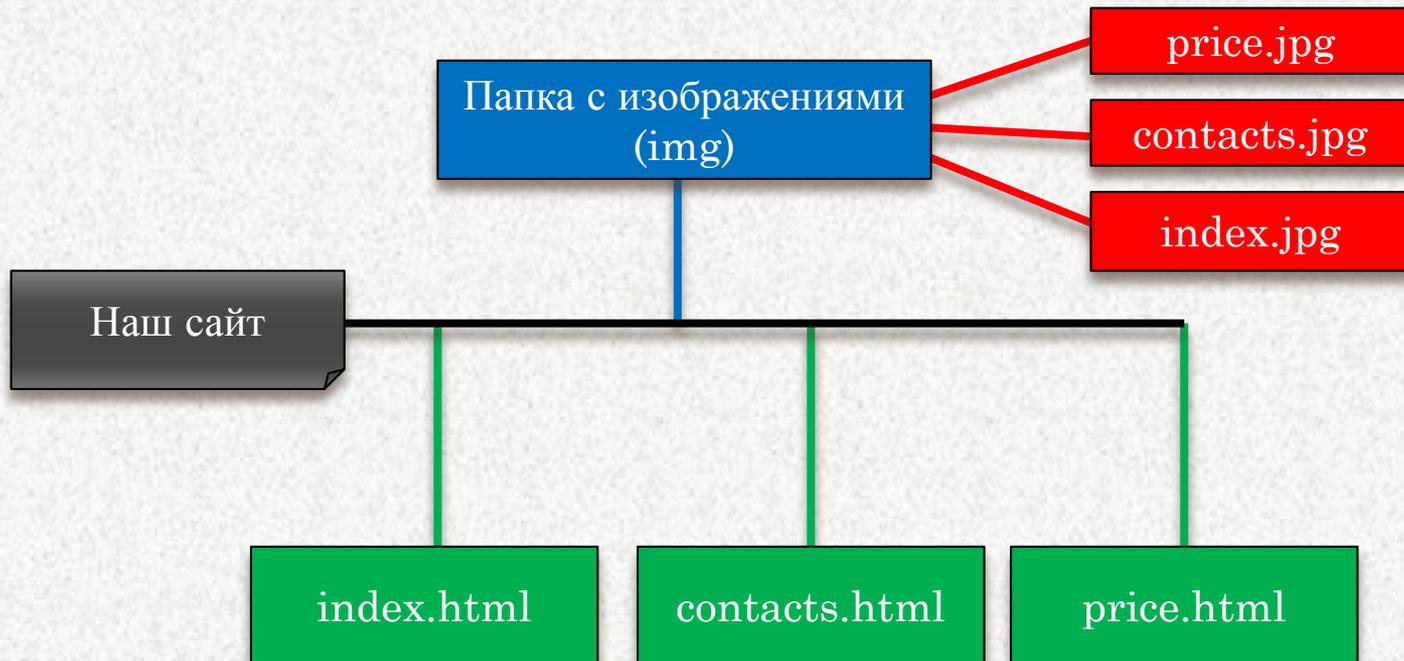
Минусы - Когда появится необходимость в смене домена(имени сайта) или например при переносе сайта с локальной машины на сервер.

Плюсы - Кража контента с вашего сайта, при использовании абсолютных путей, можно получить обратные ссылки с сайта, на котором находится сворованный контент.

Относительный путь

`src="images/povsednevniy-etiket/etiket-na-ulitse1.jpg"`
`src="../images/povsednevniy-etiket/etiket-na-ulitse1.jpg"`

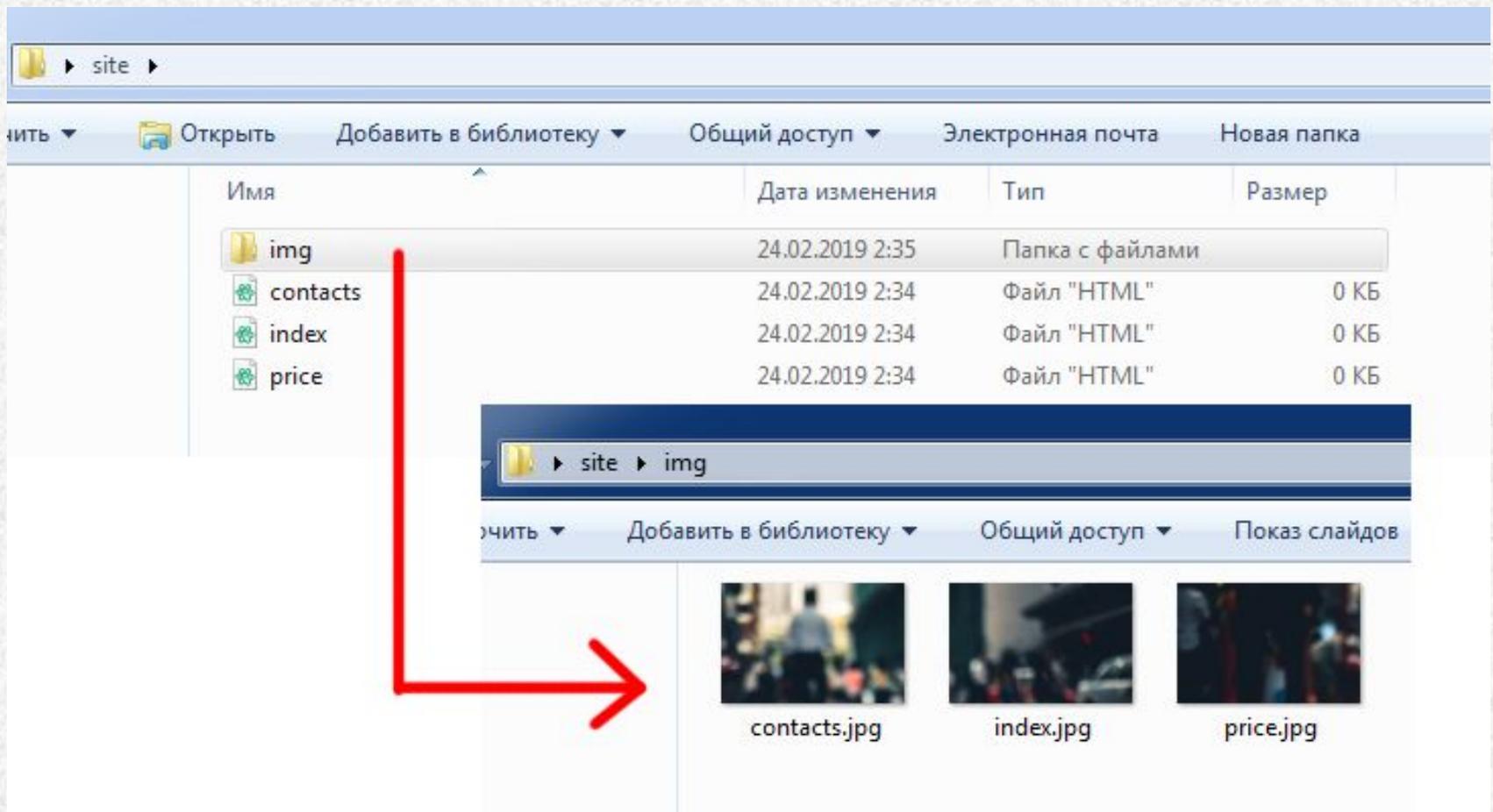
В основном относительный путь указывается тогда, когда Вам нужно отправить посетителя на другую страницу Вашего сайта, или вставить объект (например изображение) на одной из страниц



Путь относительно документа

`src= "img/contacts.jpg"`

В основном такой путь используется тогда, когда текущий и связанный документ(страница) находятся в одной директории.

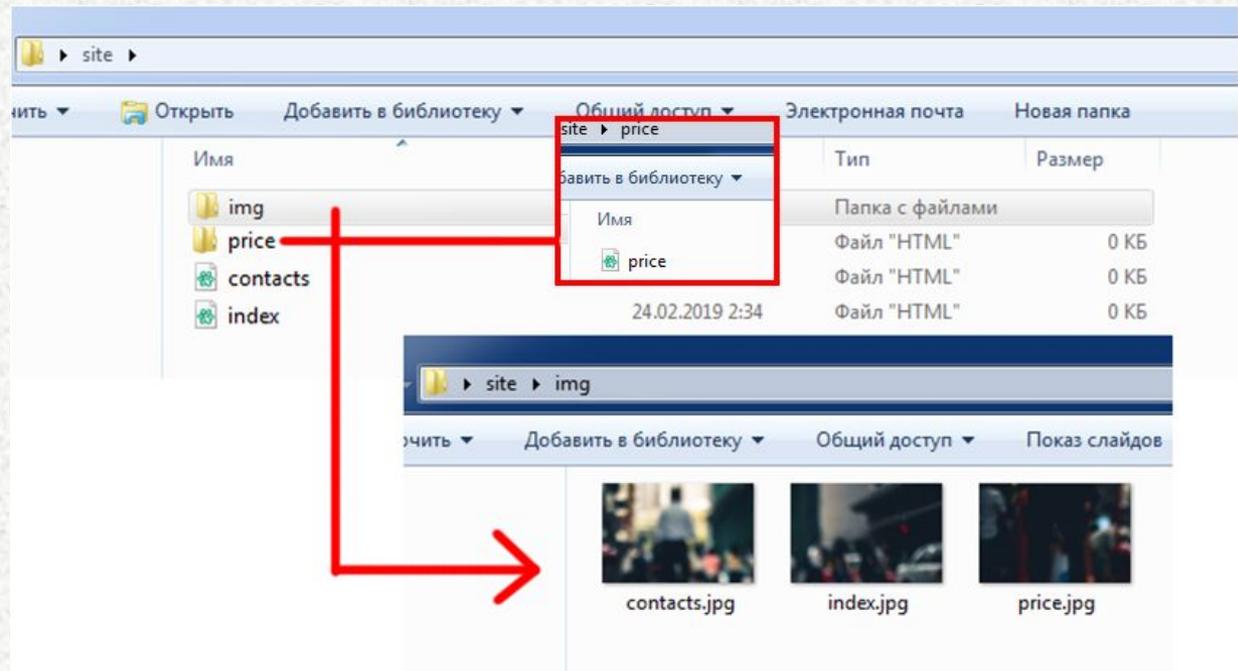


Путь относительно корня сайта

`src= "../img/price.jpg"`

`src= "/img/price.jpg"`

../ означает переход на одну директорию(уровень) выше в иерархии каталогов, символ **/** обозначает переход на один уровень ниже



Самый простой способ определить корневой относительный путь — взять абсолютный и отбросить `http://` и имя хоста.

`https://etikket.ru/images/povsednevnyi-etiket/etiket-v-lifte.jpg`

`src= "/images/povsednevnyi-etiket/etiket-v-lifte.jpg"`

Тег alt=""

```

```

Альтернативное название картинки если вдруг картинка не прогрузилась, или пользователь выключил показ изображений, или Вы не верно указали адрес изображения.

Служит для текстового описания того что изображено на картинке.

В seo оптимизации играет важную роль.

Размер картинки width, height

```

```

```

```

Без задания размеров изображения 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 class для изображения

```

```

Тег class служит для стилизации изображения, задания максимального размера или просто размера, обрамления рамкой, обтекания текстом и т.д.

```

```

```
img.full  
{  
width: 200px;  
height: 200px;  
}
```

Тег class для изображения

```

```

Тег **class** служит для стилизации изображения, задания максимального размера или просто размера, обрамления рамкой, обтекания текстом и т.д.

```
img.full  
{  
border-color: red;  
border-width: 8px;  
border-style: dotted;  
}
```

ИЛИ

```
img.full  
{  
border: dotted 8px red;  
border: 8px dotted red;  
border: red dotted 8px;  
border: red 8px dotted;  
}
```

Свойство `border` служит для обрамления рамкой любого элемента на сайте

```
img.full
{
border-bottom
border-top
border-left
border-right
}
```

ИЛИ

```
img.full
{
border: dotted 8px red;
border: 8px dotted red;
border: red dotted 8px;
border: red 8px dotted;
}
```

```
img.full:hover
{
border-bottom
border-top
border-left
border-right
}
```

ИЛИ

```
img.full:hover
{
border: dotted 11px green;
border: 8px dotted green;
border: green dotted 8px;
border: green 8px dotted;
}
```

Свойство `border` служит для обрамления рамкой любого элемента на сайте

```
img.full  
{  
border-radius}
```

Свойства `margin` и `padding` для изображения

```

```

Тег `class` служит для стилизации изображения, задания максимального размера или просто размера, обрамления рамкой, обтекания текстом и т.д.

```

```

```
img.full  
{  
width: 200px;  
height: 200px;  
}
```

Выравнивание изображения относительно текста или картинки

```

```

`align="left";`

left Выравнивает изображение по левому краю окна.

right Выравнивает изображение по правому краю окна.

Выравнивание изображения относительно текста или картинки

```

```

```
img.full  
{float:left;  
float:right;}
```

Отмена выравнивания

clear: none | left | right | both | inherit

None	Отменяет действие свойства clear, при этом обтекание элемента происходит, как задано с помощью свойства float или других настроек.
Both	Отменяет обтекание элемента одновременно с правого и левого края. Это значение рекомендуется устанавливать, когда требуется снять обтекание элемента, но неизвестно точно с какой стороны.
Left	Отменяет обтекание с левого края элемента. При этом все другие элементы на этой стороне будут опущены вниз, и располагаться под текущим элементом.
Right	Отменяет обтекание с правой стороны элемента.
Inherit	Устанавливает значение родителя.

Отмена выравнивания

html –

```
<h2>Тестовая картинка 1</h2>
```

```
<div class="left">
```

```
</div>
```

```
<p>небольшой текст</p>
```

```
<h2>Тестовая картинка 2</h2>
```

```
<div class="left"></div>
```

```
<p> небольшой текст </p>
```

Css -

```
.left {  
  float: left;  
}
```

Отмена выравнивания

html –

```
<h2>Тестовая картинка 1</h2>
```

```
<div class="left">
```

```
</div>
```

```
<p>небольшой текст</p>
```

```
<div style="clear:both"></div>
```

```
<h2>Тестовая картинка 2</h2>
```

```
<div class="left"></div>
```

```
<p> небольшой текст </p>
```

Css -

```
.left {  
  float: left;  
}
```

Фон страницы – свойство
background:

Задание фона в виде цвета: background-color

Html –

```
<div class="one"> bla bla bla text </div>
```

Css –

```
.one {  
  background-color: red;  
  padding: 20px;  
  color: white;  
  font-size: 18px;  
  font-weight: bold;  
}
```

```
.one:hover {  
  background-color: black;  
}
```

Обязательное задание фона для элемента <body>

```
body {  
  background-color: #e2e2e2;  
  width: 100%;  
  margin: 0;  
  min-height: 500px;  
}
```

```
body:hover {  
  background-color: white;  
}
```

Задание фона в виде изображения background-image

Свойство устанавливает изображение в качестве фона для элемента.

Фоновым изображением может быть картинка или градиент, который задаётся с помощью функций

background-image: linear-gradient(),

background-image: radial-gradient()

или с помощью функций повтора градиента

background-image: repeating-linear-gradient(),

background-image: repeating-radial-gradient()

background: linear-gradient(to bottom left, red 10%, blue 20%);

background: radial-gradient(at left top, #1800ff, #ff0000);

background-image: linear-gradient(to top left, red, blue);

Задание фона в виде изображения background-image

Свойство устанавливает изображение в качестве фона для элемента.

```
body {  
background-image: url(https://orehinfo.ru/images/back1.jpg);  
width: 100%;  
margin: 0;  
min-height: 12000px;  
}
```

Свойство **background-repeat** определяет, каким образом будет повторяться фоновый рисунок.

repeat	Весь фон страницы будет заполнен фоновым рисунком. Если при этом задать background-position, то повтор будет осуществляться с указанной позиции. Значение по умолчанию.
no-repeat	Фоновое изображение не будет повторяться.
repeat-x	Фоновый рисунок повторяется от левого до правого края веб-страницы по верхнему краю страницы.
repeat-y	Фоновый рисунок повторяется от верхнего до нижнего края веб-страницы по левому краю страницы.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.
round	Фоновое изображение не обрезается по границе элемента ни по вертикали, ни по горизонтали. Использование значения space приводит к тому, что изображение заполняет с промежутками все пространство элемента по ширине и высоте в целом виде, без обрезки.
space	Значение round приводит к масштабированию фонового изображения по ширине и высоте так, чтобы оно заполняло все пространство элемента целыми частями без обрезки оставляя пустое пространство

Свойство `background-position` определяет начальную позицию фонового изображения в виде горизонтальной и вертикальной координат посредством ключевых слов, точных абсолютных и процентных значений. Значение по умолчанию `background-position: 0% 0%`

left top,
left center,
left bottom,
right top,
right center,
right bottom,
center top,
center center,
center bottom

Позиционирование по горизонтали и вертикали задаётся с помощью пары ключевых слов. Если задано одно ключевое слово, второе примет значение center.

px / em / %

Указывается два значения, первое определяет расстояние между левой стороной изображения и левым краем элемента-контейнера (по горизонтали), второе значение указывает расстояние между верхней стороной изображения и верхним краем элемента-контейнера (по вертикали). Также можно использовать отрицательные значения для обрезки части изображения.

Процентное значение рассчитывается относительно самого изображения и относительно элемента-контейнера, в котором оно расположено. Например, при задании `background-position:50% 50%`; центр изображения совпадет с центром элемента-контейнера. Одновременно можно комбинировать значения в px, em и %.

initial

Устанавливает значение свойства в значение по умолчанию.

inherit

Наследует значение свойства от родительского элемента.

Также благодаря свойству позиционирования, для одного блока можно использовать несколько фоновых изображений

```
background-image: url(http://studmebel.com.ua/wp-content/uploads/2014/02/ClipArt_26.jpg), url(https://i.pinimg.com/originals/83/08/1e/83081e1....png),  
url(http://www.clipartbest.com/cliparts/Kcj/grG/KcjgrGXni.png);  
background-repeat: repeat-x;  
background-position: 0 309px, 0 198px, 0 99px;  
background-size: 5%;
```

Свойство `background-attachment` позволяет фиксировать фоновое изображение при прокрутке страницы.

<code>scroll</code>	Фоновое изображение прокручивается вместе с текстом и другим содержимым. Значение по умолчанию.
<code>fixed</code>	Предотвращает перемещение, фиксирует фоновое изображение на заднем плане.
<code>local</code>	Фоновое изображение прокручивается вместе с содержимым элемента.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

[Ссылка на html](#)

[Ссылка на css](#)

Свойство **background-size** позволяет масштабировать фоновое изображение по вертикали и горизонтали описывает, как изображение будет растягиваться и обрезаться, чтобы полностью закрыть собой фоновую область.

С помощью этого свойства изображение также можно уменьшать по ширине и по высоте. **background-size: 300px 150px;**

auto	Значение по умолчанию. Высота и ширина изображения равны его оригинальным размерам.
px / em /cm	Размер задается парой значений, первое значение устанавливает ширину изображения, второе — высоту. Для того, чтобы фон масштабировался вместе с текстом, размеры изображения нужно задавать в em.
%	Задаёт размер фонового изображения в процентах от ширины или высоты элемента, которое заполняется фоном.
cover	Масштабирует изображение с сохранением пропорций так, чтобы его ширина или высота равнялась ширине или высоте блока.
contain	Масштабирует изображение с сохранением пропорций таким образом, чтобы оно целиком поместилось внутри блока.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

**Спрайты: меньше картинок –
больше скорость**



```
<div class="yes-no"></div>
```

Css

```
.yes-no{  
  background:  
url(https://basicuse.net/var/plain_site/storage/images/media/images/articles/pr  
og_lang/textile/html_css/getting_started_with_bootstrap/glyphicons/61850-1-en  
g-GB/glyphicons_fit_content_width.png) -163px 243px;  
  width: 56px;  
  height: 48px;  
}
```

```
.yes-no:hover{  
  background:  
url(https://basicuse.net/var/plain_site/storage/images/media/images/articles/pr  
og_lang/textile/html_css/getting_started_with_bootstrap/glyphicons/61850-1-en  
g-GB/glyphicons_fit_content_width.png) -240px 242px;  
  width: 56px;  
  height: 48px;  
}
```

Сайты для создания спрайтов

<https://www.toptal.com/developers/css/sprite-generator>

<https://spritegen.website-performance.org/>

HTML5 – конкурент Flash

Вставка видео на странице посредством тега **<video>**

```
<video  
src="https://html5book.ru/examples/media/martynko.  
mp4" controls  
poster="https://avatars.mds.yandex.net/get-pdb/81227  
1/f3675e7a-5724-4a0a-accd-b73b22d3661b/s1200"  
type="video/mp4" width="640px"  
height="350px"></video>
```

Атрибут **controls** отвечает за появление элементов управления видеоплеером. Вы можете добавить изображение с помощью атрибута **poster**, которое браузер будет использовать, пока загружается видео или пока пользователь не нажмет на кнопку воспроизведения, а также задать высоту и ширину видео.

Как и в случае с аудиофайлами, рекомендуется перечислять в **<source>** все форматы, начиная с более предпочтительного. Также нужно указывать MIME-тип для каждого видеофайла.

Атрибут	Описание, принимаемое значение
autoplay	Автоматическое воспроизведение видеоофайла сразу же после загрузки страницы.
controls	Указывает браузеру, что нужно отобразить базовые элементы управления воспроизведением (воспроизведение, пауза, громкость).
height	Задает высоту окна для отображения видеоданных, возможные значения: px или %
loop	Циклическое воспроизведение видеофайла.
muted	Выключает звук при воспроизведении видеофайла.
poster	URL файла изображения, которое будет отображаться во время загрузки видеофайла или до тех пор, пока пользователь не нажмет на кнопку PLAY. Если атрибут не задан, то будет отображаться первый кадр видеофайла.
preload	<p>Атрибут, отвечающий за предварительную загрузку видеоконтента. Не является обязательным, некоторые браузеры игнорируют его.</p> <p>Возможные значения:</p> <p>auto — браузер загружает видеофайл полностью, чтобы он был доступен, когда пользователь начнет его воспроизведение.</p> <p>metadata — браузер загружает первую небольшую часть видеофайла, чтобы определить его основные характеристики.</p> <p>none — отсутствие автоматической загрузки видеофайла.</p>
src	Содержит абсолютный или относительный URL-адрес видеофайла.
width	Задает ширину окна для отображения видеоданных, возможные значения: px или %

```
<video controls width="400" height="300">  
<source src="video.mp4" type="video/mp4">  
<!-- MP4 для Safari, IE9, iPhone, iPad, Android, и Windows Phone 7 -->
```

```
<source src="video.webm" type="video/webm">  
<!-- WebM/VP8 для Firefox4, Opera, и Chrome -->
```

```
<source src="video.ogv" type="video/ogg">  
<!-- Ogg/Vorbis для старых версий браузеров Firefox и Opera -->
```

```
<object data="video.swf"  
type="application/x-shockwave-flash">  
<!-- добавляем видеоконтент для устаревших браузеров, в которых нет поддержки  
элемента video -->  
<param name="movie" value="video.swf">  
</object> </video>
```

```
<video controls width="400" height="300">  
<source src="video.mp4" type="video/mp4">  
<source src="video.webm" type="video/webm">  
<source src="video.ogv" type="video/ogg">  
<object data="video.swf"  
type="application/x-shockwave-flash">  
<param name="movie" value="video.swf">  
</object> </video>
```

Элемент `<source>` используется для указания нескольких медиа-ресурсов для `<audio>` и `<video>`. Добавляет альтернативные видео/аудио файлы, которые браузер может выбрать из предложенных на основании поддерживаемого им типа носителя или кодека.

`src` - Содержит абсолютный или относительный URL-адрес медиафайла.

`type` – Определяет тип медиафайла.

На данный момент браузеры поддерживают три основных видео формата:

Формат	Видеокодек	Аудиокодек
.mp4	H.264	AAC
.ogg/.ogv	Theora	Vorbis
.webm	VP8	Vorbis

Видео в формате .avi на сайте средствами HTML5 не воспроизводится. Поэтому его необходимо перекодировать в эти три формата с соответствующими видео и аудиокодеками для вывода на сайте. Для этого можно использовать видеоконвертеры.

Если вы хотите выровнять видеоплеер на странице, нужно обернуть элемент `<video>` в контейнер `<div>` с присвоенным классом, для которого задаются ширина и высота, соответствующие размерам вашего видео. Далее, с помощью CSS-свойств можно задать отступы, выравнивание на странице и т.д.

Вставка аудио на странице посредством тега **<audio>**

HTML5-аудио предоставляет улучшенные возможности работы с аудио контентом. До недавнего времени единственным способом добавления звуковых файлов на веб-страницы было интегрирование фонового звука с помощью тега **<bgsound>**, который проигрывался во время просмотра пользователем страницы без возможности выключения.

HTML5-элемент `<audio>` используется для внедрения звукового контента в веб-страницы. В общем виде HTML-разметка имеет следующий вид:

```
<audio src="name.ogg" controls></audio>
```

Атрибут `controls` добавляет отображение браузерами интерфейса управления аудио плеера — кнопки воспроизведения, паузы, громкости.

HTML5-элемент `<audio>` используется для внедрения звукового контента в веб-страницы. В общем виде HTML-разметка имеет следующий вид:

```
<audio src="name.ogg" controls></audio>
```

Атрибут `controls` добавляет отображение браузерами интерфейса управления аудио плеера — кнопки воспроизведения, паузы, громкости.

В настоящий момент не существует аудио формата, который бы работал во всех браузерах, поэтому для обеспечения доступности контента максимально широкой аудитории рекомендуется включать несколько источников звука, представленных с использованием атрибута `src` элемента `<source>`.

Одновременно можно добавить резервный контент для браузеров, которые не поддерживают элемент `<audio>`

```
<audio controls>
```

```
<source src="name.ogg" type="audio/ogg">
```

```
<source src="name.mp3" type="audio/mpeg">
```

```
<a href="sounds/name.mp3">Скачать name.mp3</a>
```

```
</audio>
```

Атрибут	Описание, принимаемое значение
autoplay	Автоматическое воспроизведение аудио файла сразу же после загрузки страницы.
controls	Указывает браузеру, что нужно отобразить базовые элементы управления воспроизведением (начинать и останавливать воспроизведение, переходить в другое место записи, регулировать громкость).
loop	Циклическое воспроизведение аудио файла.
muted	Выключает звук при воспроизведении аудио файла.
preload	Атрибут, отвечающий за предварительную загрузку аудио контента. Не является обязательным, некоторые браузеры игнорируют его. Возможные значения: auto — браузер загружает аудио файл полностью, чтобы он был доступен, когда пользователь начнет его воспроизведение. metadata — браузер загружает первую небольшую часть аудио файла, чтобы определить его основные характеристики. none — отсутствие автоматической загрузки аудио файла.
src	Содержит абсолютный или относительный URL-адрес аудио файла.

Создание изображений и анимации посредством тега
<canvas>

<https://www.youtube.com/watch?v=4P0i8Q1Q6fE>

Использование svg формата

Хоть формат не особо новый, но свою популярность заслужил благодаря внедрению его на сайты.

Все последние версии браузеров уже получили поддержку данного формата.

Использование `svg` формата

SVG – Scalable Vector Graphics (Масштабируемая Векторная Графика) – основанный на векторе формат для веб-разработчиков.

Данный тип формата стал столь популярен благодаря тому, что его изображения возможно отображать на больших разрешениях без потери качества, так как **SVG** – это векторный формат.

SVG имеет небольшой вес по сравнению с другими форматами.

У этого формата просто невероятные возможности.

Флаги, символы, элементы интерфейса. И это только небольшой список того, как его можно использовать.

Самый большой плюс его использования, это то что он является векторным форматом, то есть его возможно использовать на любых дисплеях – каким бы ни было их разрешение, он везде будет отображаться одинаково.

Плюсы использования SVG

Так почему же стоит отказаться от привычных нам JPG или GIF в пользу SVG? Есть достаточно много причин того, почему же все таки стоит его использовать в вашем проекте.

- SVG – векторный формат, именно поэтому его стоит его использовать для адаптивных сайтов, на которых размер изображений зависит от разрешения дисплея пользователя, и SVG безупречно соответствует всем требованиям.
- SVG изображения используют XML для определения его свойств, и, соответственно, у него есть способность к еще большему сжатию.
- SVG изображениями очень просто управлять, и это открывает еще больше возможностей для дизайнеров, которые могут менять цвет, добавлять тень, фильтры, размытие и еще много других эффектов.
- SVG довольно прост для понимания
- SVG работает с открытыми веб стандартами
- Вы можете использовать обычный текстовый редактор кода для создания SVG. Это дает вам некоторую свободу в действиях, все зависит лишь от ваших потребностей и от уровня экспертных знаний в данной области.

Минусы использования SVG

Вы вероятно потрясены тем, сколько же у SVG плюсов. И возможно подумали, что у него нет минусов. Но нет, парочка точно есть, и вот некоторые из них:

- Нет поддержки у старых браузеров, таких как Explorer 8 и ниже.
- SVG не может использоваться в фотографиях, так как это векторный формат, и применяется он для создания различных фигур и линий.

Замена jpg, gif, png на svg

https://doromarine.kz/media/mod_languages/images/kz.svg

https://doromarine.kz/media/mod_languages/images/kz.gif

С помощью `svg` можно рисовать

Размеры и координаты задаются для разных фигур по-разному, а вот фон и обводка — одинаковы для всех:

`fill` — заливка. Можно задать цвет любым способом.

Прозрачность — `none` или `transparent`.

Цвет по умолчанию — черный;

`stroke` — цвет обводки;

`stroke-width` — толщина обводки, по умолчанию — 1.

Без `stroke` не работает.

```
<svg width="230" height="140">
```

```
<rect x="5" y="5" width="220" height="130" fill="skyblue" stroke="steelblue"  
stroke-width="5" />
```

```
</svg>
```

`x, y` — координаты левого верхнего угла фигуры;

`width, height` — ширина и высота прямоугольника.

Закругленные уголки задаются параметрами `rx` и `ry`. Если задан только один из параметров, скругление по вертикали и по горизонтали будет одинаковым:

```
<svg width="230" height="140"> <rect x="5" y="5" width="220" height="130" rx="20" fill="yellowgreen" stroke="green" stroke-width="5" /> </svg>
```

Если заданы оба параметра (и они таки разные), радиус скругления будет разным:

```
<svg width="230" height="140"> <rect x="5" y="5" width="220" height="130" rx="10" ry="50" fill="khaki" stroke="darkkhaki" stroke-width="5" /> </svg>
```

Много угольники рисуются с помощью - polygon

```
<svg width="230" height="140">  
<polygon points="5,135 115,5 225,135" fill="violet" stroke="purple"  
stroke-width="5" />  
</svg>
```

Или

```
<svg width="140" height="130">  
<polygon points="70,5 90,41 136,48 103,80 111,126 70,105 29,126 36,80 5,48  
48,41" fill="turquoise" stroke="lightseagreen" stroke-width="5" />  
</svg>
```

В points задаются x,y-координаты вершин фигуры, через пробел.
polygon замыкается сам по себе, последнюю точку можно не указывать.

Круги рисуются с помощью `circle`

```
<svg width="140" height="140">  
<circle r="65" cx="70" cy="70" fill="orangered" stroke="crimson"  
stroke-width="5" />  
</svg>
```

`r` — радиус круга; `cx`, `cy` — координаты центра круга.

Эллипс с помощью `ellipse`

```
<svg width="230" height="140">  
<ellipse rx="110" ry="60" cx="115" cy="70" fill="gold" stroke="orange"  
stroke-width="5" />  
</svg>
```

`rx`, `ry` — горизонтальный и вертикальный радиусы эллипса; `cx`, `cy` — координаты центра эллипса.

Более сложные фигуры можно сделать из сочетания простых:

При этом стиль фигур можно задать один раз для всей группы, а не отдельно для каждой фигуры:

```
<svg width="250" height="210" style="border: 1px solid">
```

```
<g fill="#333">
```

```
<ellipse rx="45" ry="37" cx="55" cy="60" transform="rotate(-45, 55, 55)"/>
```

```
<ellipse rx="45" ry="37" cx="190" cy="60" transform="rotate(45, 190, 55)"/>
```

```
<circle r="67" cx="122" cy="125"/>
```

```
</g>
```

```
</svg>
```

Так же можно рисовать линии с помощью **line**

```
<svg width="230" height="140">  
<line x1="220" y1="10" x2="20" y2="130" stroke="violet" stroke-width="5" />  
</svg>
```

x1, y1, x2, y2 — координаты начала и конца линии.

Или линии соединенные между собой - **Polyline**

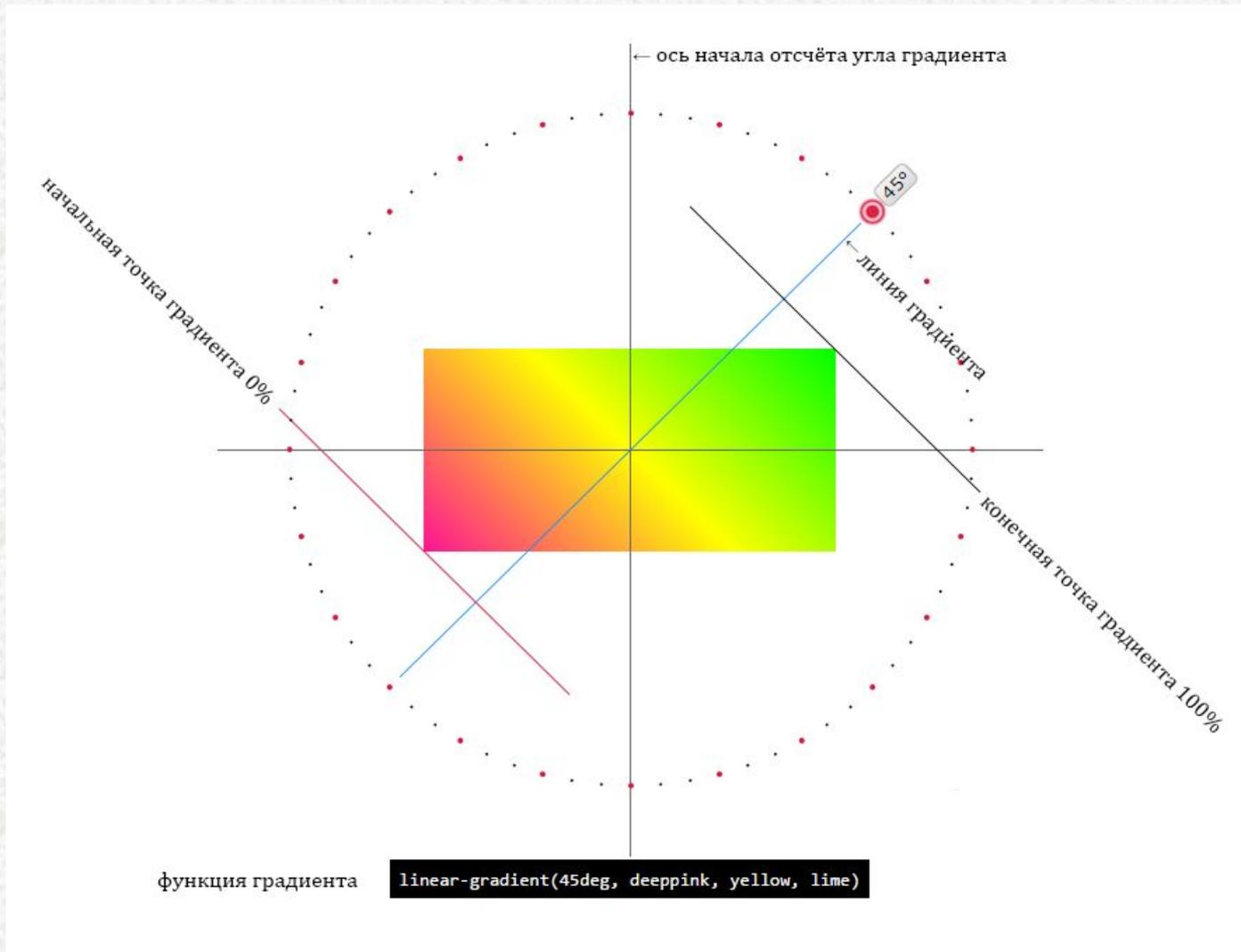
```
<svg width="260" height="140">  
<polyline points="5,135 30,5 55,135 80,5 105,135 130,5 155,135 180,5 205,135  
230,5 255,135" fill="none" stroke="orangered" stroke-width="5" />  
</svg>
```

В points задаются x,y-координаты точек, каждая точка через пробел.

Новые свойства: CSS3

Работа с фоном: создание градиентов

{background: linear-gradient(угол / сторона или угол наклона с помощью ключевого слова (пары ключевых слов), первый цвет, второй цвет и т.д.);}



Направление градиента может быть задано двумя способами:

с помощью угла наклона в градусах `deg`, значение которого определяет угол наклона линии внутри элемента.

```
div { height: 200px; background: linear-gradient(45deg, #EECFBA, #C5DDE8); }
```

с помощью ключевых слов `to top`, `to right`, `to bottom`, `to left`, которые соответствуют углу градиента, равному `0deg`, `90deg`, `180deg` и `270deg` соответственно.

```
div { height: 200px; background: linear-gradient(to right, #F6EFD2, #CEAD78); }
```

Для неравномерного распределения цветов указывается начальная позиция каждого цвета через точки остановки градиента, так называемые **color stops**.

Точки остановки задаются в %, где `0%` — начальная точка, `100%` — конечная точка, например:

```
div { height: 200px; background: linear-gradient(to top, #E4AF9D 20%, #E4E4D8 50%, #A19887 80%); }
```

Радиальный градиент `radial-gradient()`

Радиальный градиент отличается от линейного тем, что цвета выходят из одной точки (центра градиента) и равномерно распределяются наружу, рисуя форму круга или эллипса.

```
{background: radial-gradient(форма градиента / размер / позиция центра, первый цвет, второй цвет и т.д.);}
```

Форма градиента определяется ключевыми словами `circle` или `ellipse`. Если форма не задана, по умолчанию радиальный градиент принимает форму эллипса.

```
div { height: 200px; background: radial-gradient(white, #FFA9A1); }
```

Позиция центра задаётся с помощью ключевых слов, используемых в свойстве `background-position`, с добавлением приставки `at`.

Если позиция центра не задана, используется значение по умолчанию `at center`.

```
div { height: 200px; background: radial-gradient(at top, #FEFFFF, #A7CECC); }
```

С помощью пары значений, указанных в единицах длины `%`, `em` или `px`, можно управлять размером эллипсообразного градиента.

Первое значение задает ширину эллипса, второе — высоту.

```
div { height: 200px; background: radial-gradient(40% 50%, #FAECD5, #CAE4D8); }
```

Генераторы css градиентов

<https://yandex.kz/search/?text=css%20%D0%B3%D1%80%D0%B0%D0%B4%D0%B8%D0%B5%D0%BD%D1%82%20%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80&lr=162>

Работа с границами: скругленные края у блоков – свойства `border-radius`

`border-radius`: 100%;

`border-radius`: 100px;

Задаёт скругление краев у элементов

Задание полупрозрачности элементам страниц – свойство `opacity`

Свойство `opacity` задает прозрачность любому элементу к примеру

`opacity: 0.5` – задаст прозрачность в половину

Чем меньше значение тем более прозрачным будет объект

При значении 0 прозрачность будет максимальной, и объект исчезнет. Значение 1 стоит по умолчанию.

Домашнее задание

На своем сайте добавить как минимум 3 картинки в текст
И добавить фон сайта

Гиперссылки. Принципы навигации web-сайта

Общие сведения о гиперссылках

Гиперссылка (англ. *hyperlink*) —

часть гипертекстового документа, ссылающаяся на другой элемент (команда, текст, заголовок, примечание, изображение) в самом документе, на другой объект (файл, каталог, приложение), расположенный на локальном диске или в компьютерной сети, либо на элементы этого объекта.

Гиперссылка — фрагмент HTML-документа и его базовый элемент

[Ссылка на следующую страницу](#)

Гиперссылка может быть добавлена к любому элементу гипертекстового документа и обычно выделяется визуально.

В HTML-документах текстовые ссылки по умолчанию выделяются синим цветом, при наведении на них курсором мыши в окне браузера изменяются, например, меняют цвет или выделяются подчеркиванием.

При навигации в браузере с помощью клавиатуры текстовые и графические ссылки выделяются прямоугольной пунктирной рамкой. Посещенная ранее ссылка обычно выделяется цветом, отличным от цвета не посещённой

ССЫЛКИ

«Битой» ссылкой называют такую гиперссылку, которая ссылается на отсутствующий по каким-либо причинам объект, например, если документ или файл удален или перемещен администратором ресурса, на котором он был расположен, или если сам ресурс недоступен. Обычно в таком случае на странице появляется сообщение с кодом ошибки.

[Битая ссылка](#)

Для определения ссылки в HTML используется тег `<a>`, структура которого имеет вид `Текст ссылки`,

где `filename` — имя файла или адрес в Интернете, на который необходимо сослаться, а `Текст ссылки` — текст гипертекстовой ссылки, который будет непосредственно показан в HTML-документе.

Также можно использовать аргумент `TARGET`, который показывает, в каком окне будет открыта ссылка (значения — `_blank` (новом), `_self` (текущем)).
`target="имя окна">`.

`Мои работы` — ссылается на документ `my-photo.html` в текущем каталоге, образуя гипертекстовую ссылку в виде слова «Мои работы»;
`Мой фотоальбом` — ссылается на файл `my-photo.html`, расположенный в каталоге `photo` корневого каталоге и образует ссылку в виде текста «Мой фотоальбом»;
`Мой сайт` — ссылается на ресурс, расположенный на удаленном сервере.
`Открыть в новом окне`

Человекопонятный URL = ЧПУ

URL-путь, состоящий из понятных слов, вместо идентификаторов, и отражающий файловую структуру сайта.

Например, вместо /c14/3/97/

или

/index.php?cat=10&subcat=2&id=41

будет /product/phone/Samsung/

Технически URL может содержать только буквы латинского алфавита и некоторые символы. Все другие символы, включая русские буквы, кодируются (percent-encoding) и могут выглядеть вроде %D0%9B%D1%83%D0%BA. Поэтому, обычно в адресах используют латиницу, чтобы избежать осложнений для пользователей

Абсолютная и относительная адресация

Абсолютный путь

Ссылка на черный яндекс

Когда ссылка представляет из себя полный URL файла или страницы, это и есть абсолютный путь

Минусы – при замене домена ссылка становится не рабочей

Относительный путь

`Прайс-лист`

`Скачать прайс-лист`

В основном относительный путь указывается тогда, когда Вам нужно отправить посетителя на другую страницу Вашего сайта

/ 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

Самый простой способ добавить ссылку

1) Если ссылка внешняя то указывать ссылку полностью

`Прайс-лист`

2) Если ссылка указывается на ваш сайт

`Прайс-лист`

Ссылка на почту

```
<a href="mailto:simple-t@mail.ru?  
subject=Это письмо с сайта&Body=Новый текст для  
письма&cc=zuev@itstep.kz&bcc=phrakz@yandex.ru"  
title="Напишите нам">Напишите мне письмо</a>
```

mailto – Адрес электронной почты

?subject=Тема письма

&Body=Тело письма

&cc=Кому отправить копию

&bcc=Кому отправить скрытую копию

title=Подсказка при наведении на ссылку

Ссылка на телефон (срабатывает только на мобильных устройствах)

```
<a href="tel:+7-708-18-118-76" target="_blank" title="Позвонить"></a>
```

Ссылка на звонок в Skype

```
<a href="skype:логин?call">логин</a>
```

Ссылка на WhatsApp

```
<a href="https://api.whatsapp.com/send?phone=77081811876&text=Здравствуйтесь, у меня появились вопросы по HTML"> Написать на WhatsApp </a>
```

Атрибут name используется для определения закладки (иногда называют также якорь) внутри страницы.

Имя ссылки на закладку начинается символом #, после чего идёт название закладки

```
<p><a name="top"></a></p>
```

```
<p style="height:3000px;">Здесь много-много текста.  
Прокручивай его вниз. Или нажмите <a  
href="#bottom">Вниз</a> </p>
```

```
<p><a href="#top">Наверх</a></p>
```

```
<p><a name="bottom"></a></p>
```

В XHTML и HTML5 вместо name для определения закладки требуется указывать атрибут id.

Атрибут id — уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты. Идентификатор в коде документа должен быть в единственном экземпляре, иными словами, встречаться только один раз

```
<p><a id="top"></a></p>
```

```
<p style="height:3000px;">Здесь много-много текста. Прокручивай его вниз. Или нажмите <a href="#bottom">Вниз</a> </p>
```

```
<p><a href="#top">Наверх</a></p>
```

```
<p><a id="bottom"></a></p>
```

С помощью id можно создать ссылку на определенный элемент в тексте имеющий id

```
<p><a  
href="https://etikket.ru/etiket-v-internete/etiket-v-int  
ernete.html#h2-internet-etiket-i-bezopasnost">Ссылка  
на второй абзац статьи</a></p>
```

Скрипты работают только с id

Графические ссылки в HTML

```
<a href="#" target="_blank"></a>
```

```
<a href="#"></a>
```

```
<a href="#"></a>
```

HTML карта ссылок

Между тегами **<body></body>**

```

```

```
<map name="urlmapfoto1">
```

```
<area shape="poly" coords="636,293,589,408,684,408" title="Чистка носа" href="#green_link" data-title="Легкой движение шлефовальной машинки - и ваш носик чистый до самых костей">
```

```
<area shape="rect" coords="493,331,566,408" title="Нарумяним щеки" href="#red_link" data-title="Наш специалист заслуженный мастер спорта по волейболу с легкостью сделает Ваши щеки румяными">
```

```
<area shape="circle" coords="728,285,15" title="Линзы для глаз любого цвета" href="#blue_link" class="glaz">
```

```
</map>
```

HTML карта ссылок

```

```

```
<map name="Название вашей карты">
```

```
<area shape="Объект" coords="Координаты" title="Всплывающая подсказка" href="Ссылка" data-title="Всплывающий текст">
```

shape="poly" – Многоугольник - **coords = (x1,y1,x2,y2,...,xn,yn)**, последовательно указываются координаты **x,y** каждой точки многоугольника;

shape="rect" – Квадрат - **coords = (x1,y1,x2,y2)**, где **x1,y1** координаты левой верхней точки прямоугольника, **x2,y2** - координаты нижней правой точки прямоугольника;

shape="circle" – Круг - **coords = (x,y,r)**, где **x,y** координаты центра круга, а **r** - радиус круга;

```
</map>
```

<https://html5book.ru/images-in-html/>

HTML карта ссылок

Как вычислить координаты на изображении

Изображение должно быть фактического размера на сайте, без использования width и height!!!

Заходим в Photoshop->открываем изображение ->Жмем F8, открывается окно информации где есть указания цвета в rgb, цвета в стук и самое главное координаты по горизонтали (X) и координаты по вертикали (Y).

Навигат	Инфо	Гистогра
	R : 220 G : 190 B : 166	
	8-бит	8-бит
	X : 550 Y : 382	
	Док: 2,75M/2,75M	

Перетащите указатель мыши для взаимодействия с 3D-сценой.



Создание меню при помощи структуры списков (,)

Важно понимать что меню можно сделать и из тегов <p> и и <a> и и <div> и даже <h3-h6>

```
<ul class="navig">
<li><a href="#">Главная </a></li>
<li><a href="#">Новости </a></li>
<li><a href="#">События </a></li>
<li><a href="#">Цены </a></li>
<li><a href="#">Контакты </a></li>
</ul>
```

```
<p class="mmen"><a href="#"> Главная </a></p>
<p class="mmen"><a href="#"> Новости </a></p>
<p class="mmen"><a href="#"> События </a></p>
<p class="mmen"><a href="#"> Цены </a></p>
<p class="mmen"><a href="#"> Контакты </a></p>
```

Для каждого меню применяются свои стили. И раньше меню создавалось по средством

свойства float, **НО...**

Свойство display

Свойство `display` отвечает за вывод и визуальное отображение элементов на странице. Также с помощью свойства `display` можно изменить тип генерируемого контейнера.

Любой `html`-элемент генерирует на веб-странице прямоугольный контейнер. Все видимое на экране состоит из контейнеров разных типов.

```
<div class="art">  
<div style="background-color:red;">1</div>  
<div style="background-color:green;">2</div>  
<div style="background-color:blue;">3</div>  
<div style="background-color:yellow;">4</div>  
<div style="background-color:purple;">5</div>  
</div>
```

Свойство display

Свойство `display` отвечает за вывод и визуальное отображение элементов на странице. Также с помощью свойства `display` можно изменить тип генерируемого контейнера.

Любой `html`-элемент генерирует на веб-странице прямоугольный контейнер. Все видимое на экране состоит из контейнеров разных типов.

```
<div class="art">  
<div style="background-color:red;">1</div>  
<div style="background-color:green;">2</div>  
<div style="background-color:blue;">3</div>  
<div style="background-color:yellow;">4</div>  
<div style="background-color:purple;">5</div>  
</div>
```

По умолчанию блочные элементы генерируют структурные блоки и выводятся вертикально один над другим, занимая по ширине 100% ширины блока-контейнера.

Строковые контейнеры генерируют строковые блоки и выводятся в строке горизонтально. Ширина строковых элементов равна ширине их содержимого.

inline	Значение по умолчанию. Элемент генерирует строковый блок. Аналог — тег <code></code> .
block	Элемент генерирует структурный блок, как и тег <code><div></code> .
flex	Элемент генерирует структурный блок, который создает адаптивный контейнер для дочерних элементов. http://css.yoksel.ru/flexbox/
Особенности inline- элементов:	
- им можно задавать размеры, рамки и отступы, как и блочным элементам;	
- их ширина по умолчанию зависит от содержания, а не растягивается на всю ширину контейнера;	
inline-block	Элемент генерирует строковый блок.
inline-flex	Элемент генерирует строковый блок, который создает адаптивный контейнер для дочерних элементов.
inline-table	Элемент определяет структурный блок, который генерирует строковый блок.
list-item	Элемент генерирует структурный блок, который отображается как элемент списка <code></code> .
table	Элемент генерирует структурный блок. На странице ведет себя аналогично <code><table></code> .
table-caption	Элемент генерирует основной заголовок таблицы. На странице ведет себя аналогично <code><caption></code> .

table-column	Элемент описывает столбец ячеек, визуальное представление не генерируется. Аналог — <code><col></code> .
table-column-group	Элемент объединяет один или несколько столбцов. Аналог — <code><colgroup></code> .
table-cell	Элемент генерирует отдельную ячейку таблицы, на странице ведет себя аналогично <code><th></code> и <code><td></code> .
table-header-group	Элемент определяет группу строк заголовка, которая всегда отображается перед остальными строками и группами строк. Аналог — <code><thead></code> .
table-footer-group	Элемент определяет группу строк заголовка, которая всегда отображается после всех остальных строк и перед любым нижним основным заголовком. Ведет себя аналогично <code><tfoot></code> .
table-row-group	Элемент объединяет одну или несколько строк. Аналог — <code><tbody></code> .
table-row	Элемент является строкой ячеек. Пример — <code><tr></code> .
none	Элемент не генерирует никакой контейнер, полностью удаляясь со страницы.
inherit	Наследует свойство от родительского элемента.

Псевдоклассы

Динамические псевдоклассы

Выбирают ссылки на странице, которые имеют атрибут `href` и находятся в определенном состоянии, а также некоторые другие элементы:

`:link` — не посещенная ссылка;

`:visited` — посещенная ссылка;

`:focus` — ссылки, а также элементы форм, которые активированы посредством курсора мыши или на которые перешли с помощью клавиатуры (кнопка TAB);

`:hover` — ссылки, а также другие элементы, стили применяются при наведении пользователем на элемент;

`:active` — выбирает элемент, активированный пользователем с помощью клика мышки. Обычно применяется для ссылок, но может отбирать и другие элементы на странице.

Структурные псевдоклассы

`:first-child` — элемент, который является первым дочерним элементом некоторого другого элемента;

`:last-child` — последний дочерний элемент элемента-контейнера;

`:first-line` задает стиль первой строки форматированного текста. Длина этой строки зависит от многих факторов, таких как используемый шрифт, размер окна браузера, ширина блока, языка и т.д. В правилах стиля допустимо использовать только свойства, относящиеся к шрифту, изменению цвета текста и фона.

`:first-letter` определяет стиль первого символа в тексте элемента, к которому добавляется. К этому псевдоэлементу могут применяться только стилевые свойства, связанные со свойствами шрифта, полями, отступами, границами, цветом и фоном.

CSS свойство `cursor`

Значение

default



Пример

P {cursor: default}

crosshair



P {cursor: crosshair}

help



P {cursor: help}

move



P {cursor: move}

pointer



P {cursor: pointer}

progress



P {cursor: progress}

text



P {cursor: text}

wait



P {cursor: wait}

n-resize



P {cursor: n-resize}

ne-resize



P {cursor: ne-resize}

e-resize



P {cursor :e-resize}

se-resize



P {cursor: se-resize}

s-resize



P {cursor: s-resize}

sw-resize



P {cursor: sw-resize}

w-resize



P {cursor: w-resize}

nw-resize



P {cursor :nw-resize}

Домашнее задание

К своей странице добавить горизонтальное меню состоящее из 4 ссылок 1 ссылка на главную страницу, 2 ссылка на новую страницу, 3 и 4 ссылка просто #.

При наведении и клике на меню должен изменяться цвет ссылки. При формировании страниц обязательно использовать классы!!!

На 2 странице должна быть фотогалерея минимум из 6 фотографий

При наведении на фото должна появляться рамка во круг фото.

При нажатии фотография должна открываться в новом окне. Фон должен быть картинкой!!!