**Векторная графика** – это изображения формата *svg*, которые состоят из линий. Их можно растягивать без потери качества.

**Растровая графика** – это изображения формата *jpeg*, *png*, которые состоят из пикселей. При растягивании качество теряется, становятся видны пиксели.

В изображениях в формате *png* присутствует альфа-канал (может быть прозрачный фон или элементы).

Формат *gif* используется для коротких зацикленных анимаций. Много весит.

**Физический пиксель** – самая маленькая точка цвета, которая может быть отображена на устройстве.

**Пиксель в *CSS*** – физическая величина, которая определяет сколько места на экране будет занимать элемент. Может соответствовать х1, х2 и х3 физическому пикселю.

В зависимости от плотности пикселей на экране нужно подгружать картинки в разных размерах.

Для ускорения загрузок изображений особое внимание нужно уделить их оптимизации:

* Использовать подходящий формат. Для простых картинок это *png*, а не *jpeg*.
* Использовать нужный размер. Не нужно отдавать картинку размером 2000х2000 в блок 500х500.
* Использовать для разных разрешений экранов разные изображения.
* Оптимизировать вес изображения через сжатие в онлайн-инструментах [TinyJPG](https://tinyjpg.com/) или [Sqoosh](https://squoosh.app/).

Пока картинка грузится используют цветной фон при помощи *background-color*.

***<img>*** - одиночный тег контентного изображения. Обязательный атрибут ***src*** указывает местоположение картинки.

*<img src=”images/mobile.jpg”>*

Необязательный атрибут ***alt*** отображает текст, который пользователь видит, когда картинка не загружается:

*<img src=”images/mobile.jpg” alt=”здесь должна быть картинка, но что-то пошло не так…”>*

Необязательный атрибут ***width*** позволяет задать ширину картинки в пикселях. Это также позволит избежать рывков страницы при медленной загрузке изображений.

*<img src=”images/mobile.jpg” width=230>*

Необязательный атрибут ***srcset*** позволяет задать картинки с разной плотностью пикселей.

Необязательный атрибут ***loading=”lazy”*** позволяет подгружать картинки только при попадании на видимый экран. Пока не поддерживается всеми браузерами.

*<img src="flower.jpg"*

*srcset= "flower.jpg 1x,*

*flower-2x.jpg 2x,*

*flower-3x.jpg 3x"*

*alt="Букет ярких роз крупным планом"*

*loading=”lazy”*

*>*

***<picture></picture>*** - теги всех вариантов картинки, заданных при помощи одиночных тегов *source* и *img*. Один вариант картинки – один тег *source*. В теге *img* указываем железобетонный вариант, который точно поддерживается браузером.

*<picture>*

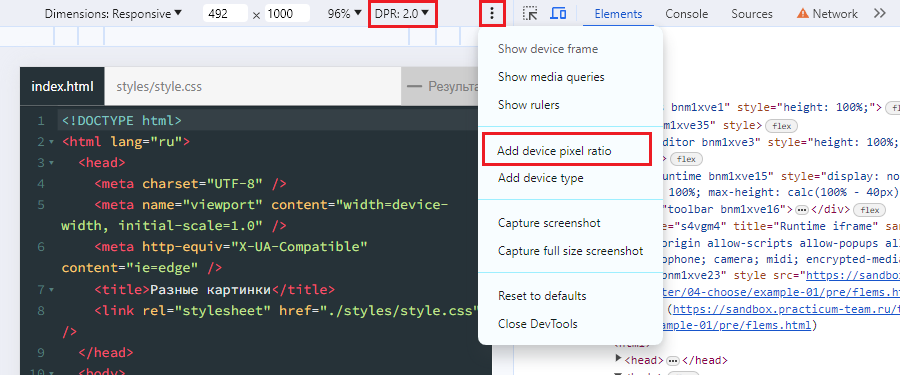
*<source srcset="flowers.avif" type="image/avif">*

*<source srcset="flowers.webp" type="image/webp">*

*<img src="flowers.jpeg" alt="Букет ярких роз крупным планом">*

*</picture>*

При тестировании кода, можно указать плотность пикселей экрана ***DRP*** в инструментах разработчика.



Функция ***image-set()*** позволяет загружать разные варианты фоновых изображений для *CSS*-свойства *background-image*. Пока не поддерживается всеми браузерами, поэтому пишется после гарантированного варианта.

*div {*

*background-image: url(flowers.jpeg);*

*background-image: -webkit-image-set( // вендорный префикс включающий*

*url("flowers.avif") type("image/avif"), // экспериментальное свойство*

*url("flowers.jpg") type("image/jpeg"));*

*background-image: image-set(*

*url("flowers.avif") type("image/avif"),*

*url("flowers.jpg") type("image/jpeg"));*

*}*

**Фрейм** – это окошко, в котором видно другую страницу сайта. Для его создания нужны теги ***iframe***. Технически фреймы ведут себя в потоке как *div*.

*<iframe src=”https://ya.ru/” frameborder=”0”></iframe>*

Атрибут ***frameborder*** со значением 0 позволяет отключить серую рамку фрейма.

Свойство ***boarder*** позволяет модифицировать или отключить серую рамку фрейма.

***Application Programming Interface*** (или ***API***) – это интерфейс взаимодействия между сайтом и сторонними программами и серверами.

Рассмотрим пример с подключением *API Youtube* через фрейм. Для этого используется ссылка вида *https://www.yotube.com/embed/идентификатор\_ролика*.

*<iframe*

*src="https://www.youtube.com/embed/RhlQvbvMg-0?start=30&mute=true" // без звука и с 30й секунды*

*title="YouTube video player"*

*frameborder="0"*

*allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"*

*allowfullscreen*

*></iframe*>

Рассмотрим пример с подключением Яндекс.Карт через фрейм. Для этого воспользуемся специальным конструктором *https://yandex.ru/map-constructor/*.

*<iframe*

*src="https://yandex.ru/map-widget/v1/?um=constructor%3A6fc6359a9fea00ada5a4032946d1df7ac9ad308d11f5e45b29cf78e2bdf6a738&amp;source=constructor"*

*width="100%"*

*height="500"*

*frameborder="0"*

*></iframe>*

Рассмотрим пример подключения своего видео без использования сторонних сервисов. Для этого нам нужны теги ***video***. Не рекомендуется позиционировать видео относительно элемента html, т.к. на некоторых размерах экранов будет неправильно высчитывать ширину в 100%, и вырвать видео из потока ***position:relative***.

Атрибут ***controls*** позволяет отобразить кнопки управления видео.

Атрибут ***muted*** позволяет выключить звук видео.

Атрибут ***autoplay*** позволяет включить автовоспроизведение видео. Поддерживается не всеми браузерами и работает в связке с ***muted***.

Атрибут ***loop*** позволяет включить зациклить видео по кругу.

Атрибут ***poster*** позволяет установить картинку, которую будет видеть пользователь, пока не запустит воспроизведение или пока видео загружается.

*<video src="video.mp4" controls muted autoplay loop poster=”poster.png”></video>*

Для подключения нескольких видео используется одиночный тег ***source***. В первую очередь нужно подключать более современные типы файлов, которые указываются в атрибуте ***type***. Браузер возьмет первый файл, который сможет проиграть, и не будет скачивать остальные файлы.

*<video controls muted loop poster=”poster.png”>*

*<source src="video.webm" type="video/webm" />*

*<source src="video.mp4" type="video/mp4" />*

*</video>*

Свойство ***object-fit*** позволяет задать как видео будет подстраиваться под заданные размеры.

*object-fit: fill; // по умолчанию, вписывается без сохранения пропорций*

*object-fit: contain; // вписывается под заданные размеры без нарушения пропорций*

*object-fit: cover; // вписывается без нарушения пропорций, обрезая все лишнее*

*object-fit: none; // отображается без изменений пропорций или размеров*

*object-fit: scale-down; // браузер сам выбирает между none и contain*

Свойство ***object-position*** позволяет выровнять изображение по осям *X* и *Y* после подгонки по размерам.

*object-position: top right; // сначала по X, потом по Y*

Рассмотрим пример подключения своего аудио без использования сторонних сервисов. Для этого нам нужны теги ***audio***. Они имеют такие же атрибуты как у тегов ***video***.

*<audio>*

*<source src="sound.mp3" type="audio/mp3" />*

*<source src="sound.ogg" type="audio/ogg" />*

*</audio>*

Как сделать свой видеоплеер можно почитать [тут](https://habr.com/ru/company/microsoft/blog/127295/).