Видео недели:

1. [Форматы графики](https://www.youtube.com/watch?v=eCgP88W2vvc)
2. [Подключение графики](https://www.youtube.com/watch?v=SMfv61OyRQc)
3. [Пути к файлам](https://youtu.be/IbdiDxWQOEw)
4. [Мой любимый цвет, мой любимый размер](https://youtu.be/MLhBxVyKTIM)
5. [Обзор редактора Figma](https://youtu.be/JKsUY9yMQnU)

**Графика**

**Форматы графики**

**Растровая**

Хранит информацию о каждом пикселе.

**JPEG или JPG**

* сжатие с потерями качества
* не поддерживает прозрачность
* подходит для многоцветных иллюстраций и фотографий

**PNG**

* сжатие без визуальной потери качества
* поддерживает прозрачность
* подходит для полупрозрачных изображений, иллюстраций, схем, иконок

**GIF**

* только 256 цветов
* поддерживает прозрачность
* только для анимированных картинок

**Векторная**

В векторных форматах хранится не само изображение, а инструкция (формула) по его построению по точкам и кривым.

**SVG**

* поддерживает прозрачность
* без потери качества при изменении размера
* возможна анимация

**Примеры различий между разными видами графики**

Здесь мы наглядно видим, что растровая графика теряет в качестве при изменении размера, а векторная остается в идеальном виде. Но зато вектором нельзя сделать сложные реалистичные фотографии.

**Как выбрать формат**

**Подключение графики**

Встроить графику в сайт можно разными способами:

* Уже знакомые нам теги **img** и **icon**
* C помощью атрибута style **background**
* Через тег **object** (на самом деле через него можно вставлять практически что угодно, и даже не только графику, но и видео). То, что указано внутри тега object, будет показываться только тогда, когда этот тег не поддерживается браузером - поэтому в этом примере там внутри обычная картинка в формате png.
* <object type="image/svg+xml" data="picture.svg">
* <img src="picture.png">
* </object>
* Специальный тег для **svg** графики

**Структура проекта**

Как только наш проект разрастается - в нем становится много страниц, картинок и прочего, стоит задуматься, а как же нам построить структуру проекта, чтобы все было просто и наглядно и сразу понятно, что где лежит (тем более, что в дальнейшем у нас добавятся еще различные элементы, стили, шрифты и т.д.) - просто добавляем новую папку *assets,* делаем в ней папку *images -* и в ней размещаем картинки:

Этот пример - не единственно возможная структура, в каждой ИТ-компании может быть принята своя, но на мой взгляд это самый простой и при этом удобный вариант, поэтому я вам даю именно его.

**Пути файлов**

Подключать графику (да и вообще любые файлы) можно тремя способами:

* **Абсолютные пути** - используют ссылку на конкретный адрес картинки на компьютере, и это будет работать только для программиста в процессе разработки. Для сервера **крайне нежелательно указывать абсолютные пути**. Примером абсолютного пути может быть: MacHD:/Users/alisa.tsvetkova/Yandex.Disk.localized/\_work/\_itgirlschool/lessons/week3/assets/images/orange-cat.png (для Mac) или C:/Users/alisa.tsvetkova/Yandex.Disk.localized/\_work/\_itgirlschool/lessons/week3/assets/images/orange-cat.png (для Windows)
* **Относительные пути** работают относительно папки, из которой этот файл вызывается. Если файл html и картинка находятся в одной и той же папке, то можно написать просто имя файла и это сработает orange-cat.png. Если картинка находится глубже в той же папке, где лежит html-файл, то можно использовать вложенную структуру папок, но все равно они будут вычисляться относительно той же папки, например: assets/images/orange-cat.png
* **Интернет-пути** указывают на файл в интернете, например <https://www.flaticon.com/svg/static/icons/svg/2111/2111646.svg>

Этот вариант отлично подходит для быстрого тестирования, но в дальнейшем лучше сохранить файл к себе и указать относительный путь к нему, потому что ссылка в интернете может в любой момент стать недействительной - и картинка исчезнет.

**Правила подключения файлов**

* Без русских имен, только английские
* Без пробелов
* Желательно без больших букв

**Как задать размер (картинки и не только)**

Размер мы задаем с помощью атрибута **style width** (ширина) и/или **style height** (высота)

* По умолчанию размер считается в пикселях *px* (минимально возможная точка экрана), это абсолютный размер:

<img src="cat.jpg" alt="the cat" style="width: 400; height: 200;>

* Но еще размер изображения можно задавать в процентах *% от родительского блока.* В этом примере картинка займет всю ширину родительского элемента:

<img src="cat.jpg" alt="the cat" style="width: 100%;>

* Помимо процентов относительными единицами являются величины **vh** (для высоты) \*\*\*\*и **vw** (для ширины)**.** Проценты вычисляются относительно размеров ближайшего родительского элемента, поэтому, если вы хотите использовать высоту и ширину окна браузера, лучше воспользоваться единицами **vh** и **vw**. По сути это % не от родительского элемента, а от окна браузера. Например, если высота окна браузера равна 900px, то 1vh будет равен 9px. Аналогично, если ширина окна браузера равна 1600px, то 1vw будет равен 16px.

<img src="cat.jpg" alt="the cat" style="width: 80vw;> - вот эта картинка займет 80% от ширины экрана браузера

**Работа с цветом**

Иногда нам не обязательно подключать целую картинку, в каких-то ситуациях достаточно просто залить какой-то блок цветом. Заливку мы делаем через атрибут **style background-color**, а цвет шрифта меняем благодаря **style color**.

Цвета в HTML могут задаваться несколькими способами:

1. **Цифро-буквенный код** **RGB** в виде шестнадцатеричного значения – используются код, заданный в шестнадцатеричной системе исчисления. Такие коды цветов в **html** состоят из трех пар шестнадцатеричных чисел. Каждая пара отвечает за насыщенность оттенка своим основным цветом:

* Первая числовая пара – отвечает за красный цвет R
* Вторая пара – за содержание зеленого цвета G
* Последняя – за содержание синего цвета B

В начале кода (*перед цифрами*) ставится решетка. Так обозначается шестнадцатеричный цветовой код. Кроме чисел от 1 до 9 в этой системе исчисления используются буквы латинского алфавита (A, B, C, D, E, F).

*Например, код белого цвета в****html****будет иметь вид* #FFFFFF

<div style="background-color: #000000; color: #FFFFFF">

здесь используется заливка черным цветом, а текст написан белым

</div>

1. **Цифро-буквенный код** **RGBA с прозрачностью** - здесь все тоже самое, только в конце добавляется еще одно значение A (alpha) для задания прозрачности. Записывается вот так *#FFFFFF****FF***

<div style="background-color: #00000022;">

здесь используется заливка черным цветом - но она частично прозрачная

</div>

1. В формате **RGB** – эта кодировка цветов в **html** основана на использовании трех значений, задаваемых в диапазоне от 0 до 255. Каждое из них определяет насыщенность оттенка одним из основных цветов:

* R – красный (**red**)
* G – зеленый (**green**)
* B – синий (**blue**)

Только, в отличие от первого варианта, мы их не переформатируем в шестнадцатеричный формат, а оставляем как есть.

<div style="background-color: rgb(255, 220, 227); color: rgb(255, 255, 255)">

а это прекрасный розовый цвет в фоне и белый в тексте

</div>

1. **RGBA** - представляет собой усовершенствованный формат **RGB**, где четвертым значением задается прозрачность цвета в виде дроби от 0 до 1 (0 - полностью прозрачный, 0.5 - наполовину прозрачный, 1 - полностью видимый)

Пример использования:

<p style="font-size:40px; background-color:rgba(100,86,143,0.2)">

background-color:rgba(100,86,143,0.2)

</p>

<p style="font-size:40px; background-color:rgba(100,86,143,0.5)">

background-color:rgba(100,86,143,0.5)

</p>

<p style="font-size:40px; background-color:rgba(100,86,143,0.8)">

background-color:rgba(100,86,143,0.8)

</p>

<p style="font-size:40px; background-color:rgba(100,86,143,1)">

background-color:rgba(100,86,143,1)

</p>

1. А еще цвет можно задавать просто словами - 'red', 'green', 'white', но конечно, такой набор слов достаточно ограничен и ими нельзя регулировать тончайшие оттенки цвета.

<div style="background-color: 'black'; color: 'white'">

здесь используется заливка черным цветом, а текст написан белым

</div>

**Графические редакторы**

**Что нужно уметь программисту**

* получение данных о тексте
* определение цвета элементов
* измерение размеров
* экспорт изображений

**Figma**

**Это первая облачная платформа для разработки интерфейсов с поддержкой командной работы в режиме реального времени.** Многие компании, занимающиеся веб-дизайном, полностью перешли на «Фигму», отказавшись от привычных Adobe Photoshop и Sketch.

Сервис обладает встроенным файловым менеджером, доступен на компьютерах и мобильных устройствах. **Одно из главных преимуществ — создание интерактивных страниц.** Заказчик открывает прототип в браузере и видит, как выглядит список при раскрытии или кнопка корзины при активации. Эти возможности экономят много времени на согласовании нюансов.

Типичные графические редакторы — мощные инструменты, которые за десятки лет развития «обросли» миллионами функций. Чтобы разобраться во всех модулях и повторять операции в автоматическом режиме, надо потратить 2-3 года на изучение и ежедневно совершенствовать знания.

С Figma **всё гораздо проще.** На освоение базовых возможностей уходит **всего несколько дней**, а нам как программистам, а не дизайнерам - **и того меньше**.

**Обзор интерфейса**

* Самый главный инструмент, который нам нужен из панели свойств - это **экспорт картинки** (логотипа, изображения, кнопки и пр.). Он находится в самом низу панели свойств. Здесь мы можем задать формат и размер выгрузки:
* И еще один очень важный функционал, который нам будет очень часто пригождаться - это просмотр свойств компонент (например, таких как размеры или вид шрифта) при клике на нее:
* Измерение цвета компонента также происходит просто по клику на него - и мы сразу видим цвет и уровень прозрачности этого цвета:

**Горячие клавиши**

**Shift + Ctrl + 4**

Включает и отключает сетку. Яркость и цвет можно изменить в настройках справа.

<https://leonardo.osnova.io/25723306-aca6-4b23-eae7-c3c59cd40053/-/resize/1200/>

**Shift + 0**

Отображает макет в 100% (масштаб 1:1)

**Ctrl + G**

Сгруппировать элементы

**Полезные инструменты для работы с графикой**

Сайты иконок и SVG-картинок

* <https://www.flaticon.com>
* <https://www.iconfinder.com>

Сайты с бесплатными фотографиями

* <https://unsplash.com>

Сайты-рисовалки прототипов интерфейсов

* <https://www.adobe.com/ru/products/xd.html>
* <https://balsamiq.cloud/> (вот этот я просто обожаю)

Сайты для компрессии (сжатия) картинок

* TinyPNG <https://tinypng.com/>
* [Compressor.io](http://Compressor.io) <https://compressor.io/>

Коды и названия цветов HTML

* <https://www.spravkaweb.ru/html/pril/color/>

**Домашнее задание на эту неделю:**

**Практическое задание 💎**

1. Доработать страницу **про Кота Великолепного** из уроков первой недели\*\*:\*\*
   * добавить картинки тремя разными способами
   * раскрасить заголовок
   * сделать блоки с разной степенью прозрачности
2. Доработать **страницу блогера**, вставив туда векторные и растровые картинки, а также анимацию:
   * Иконка
   * Логотип
   * Фото блогера
   * Картинки-фотографии в статьях
   * Анимация
3. Найти в интернете любой понравившийся макет в формате Figma и создайте страничку с **шапкой этого сайта** по этому макету (логотип, название, меню) - здесь не обязательно делать один в один, но важно сохранить структуру (очередность), картинки, цвета. Шрифты и позиционирование пока не трогайте.
4. Создайте **прототип сайта-визитки для себя** как для программиста в любом сервисе создания прототипов, например, в <https://balsamiq.cloud/> (здесь рисуем с нуля) или Figma (можно найти уже готовый пример и чуть доработать под себя, поменяв надписи и фото)
5. Проверить свои страницы валидатором и исправить ошибки (если они есть)

**Тест 💎**

1. Какой формат картинки лучше всего подходит для большого фото?
2. В чем особенность формата SVG?
3. Какой формат картинки идеально подходит для логотипа?
4. Какие форматы поддерживают анимацию?
5. Какими способами можно указывать цвет (минимум 3 варианта)?
6. Самостоятельно разберитесь, как можно задать цвет в виде градиента
7. Как можно задавать прозрачность для блока?
8. Какой способ подключения файлов самый лучший?
9. Как сделать картинку на 45% размера экрана?
10. Уменьшите размер картинок из п.2 ДЗ любым сервисом для сжатия картинок и пришлите разницу размеров До и После

**Ссылка на свою страничку из википедии 💎**

ОТВЕТЫ:

* 1. Jpeg или jpg
  2. Формат svg явл. векторной графикой, поддерживает прозрачность и анимацию, с помощью специального тега <svg> есть возможность вставить изображение в код в виде кода.
  3. png
  4. gif svg
  5. 1 способ - цифро-буквенный код RGB (#FFFFFF – белый цвет)

2 способ – кодировка цветов rgb ( rgb(265, 254, 218)

3 способ – кодировка цветов rgba, («а» отвечает за прозрачность) ( rgb(265, 254, 218, 0.5)

* 1. style="background-image: linear-gradient(to top, #FFFFFF, #000000);
  2. с помощью усовершенствованного формата rgba, где «а» отвечает за прозрачность.
  3. с помощью относительных ссылок
  4. чтобы сделать картинку на 45% размера экрана, нужно указать размеры следующим образом : style="width:45vw; height:45vh;
  5. прикладываю скрин