

HyperText Markup Language

HTML & CSS



Урок 13

Materialize

Оглавление

Materialize	3
1. Что такое Materialize? Что такое material design?	
История создания Materialize	3
2. Цели и задачи Materialize	5
3. Подключение Materialize.....	5
4. Цветовая палитра в Materialize	7
5. Система сеток в Materialize	10
6. Понятие Helper. Виды Helpers	15
7. Навигация и Materialize	19
8. Использование изображений и видео.....	23
9. Скроллинг	27
10. Пример создания сайта с помощью фреймворка Materialize	30
Домашнее задание.....	44

Материалы урока прикреплены к данному PDF-файлу. Для доступа к ним, урок необходимо открыть в программе [Adobe Acrobat Reader](#).

Materialize

1. Что такое Materialize? Что такое material design?

История создания Materialize

Materialize — это фреймворк CSS, построенный на принципах материального дизайна. Понятие material design (материальный дизайн) появилось в 2014 году благодаря компании Google. Целью Google являлось разработать дизайн, который бы учитывал унифицированный опыт пользователей на различных платформах и устройствах.

Идея material design заключается в том, что поведение и внешний вид интерфейса страницы или приложения повторяет поведение и внешний вид обычных картонных карточек, как в реальной жизни. Material design наследует плоский дизайн, в который добавлены тени, трехмерное пространство и многочисленные плавные анимации.

Примеры material design с использованием фреймворка Materialize (рис. 1).

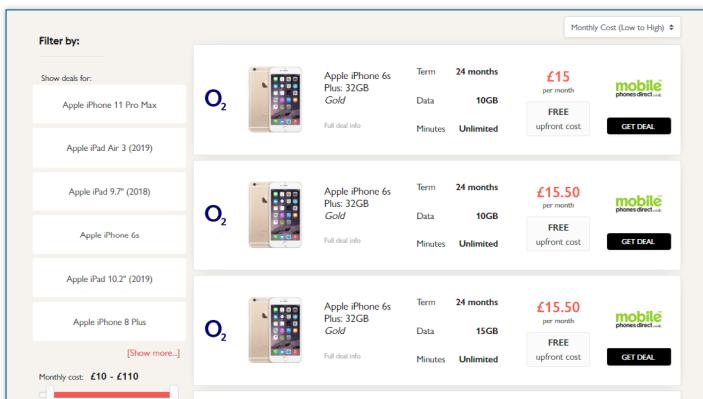


Рисунок 1. Интернет-магазин ([onecompare.com/apple](https://www.onecompare.com/apple))

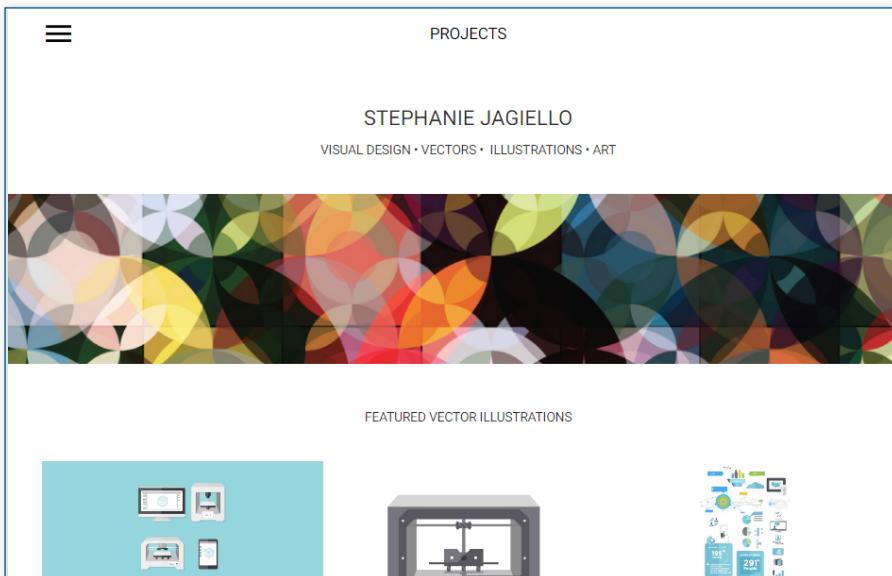


Рисунок 2. Сайт-портфолио (stephaniejagiello.com)

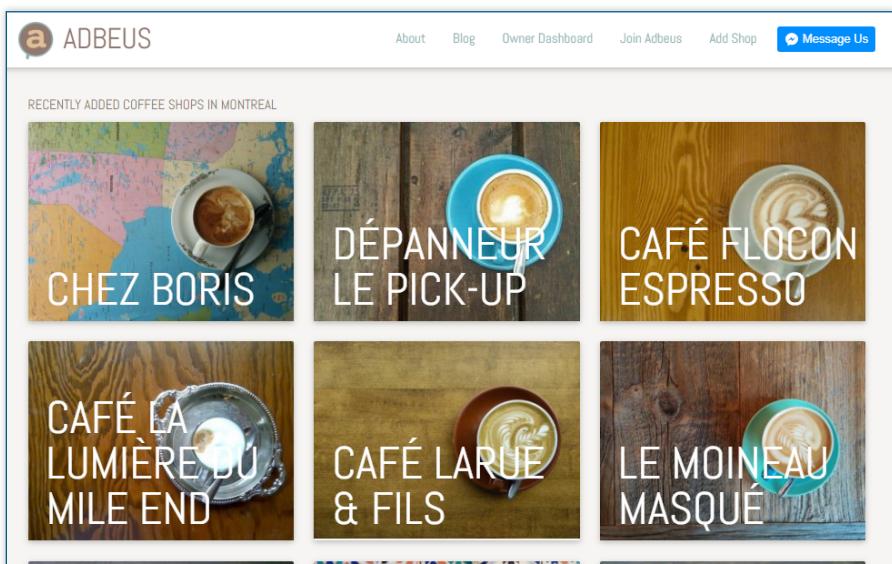


Рисунок 3. Сайт сети кафе (adbeus.com)

2. Цели и задачи Materialize

Фреймворк Materialize предназначен для простоты создания веб-страниц и приложений.

С помощью классов вы сможете:

- легко упорядочивать и компоновать элементы для отображения на различных устройствах, не прописывая дополнительные медиа-запросы;
- стилизовать элементы управления и интерфейса;
- добавлять различные анимированные элементы: всплывающие окна, аккордеоны (раскрывающиеся вкладки), выпадающие списки и так далее.

3. Подключение Materialize

Скачать Materialize можно с официального сайта, пройдя по ссылке materializecss.com/getting-started.html. После скачивания, необходимо распаковать архив в корень папки с вашим сайтом. Структура проекта должна выглядеть следующим образом:

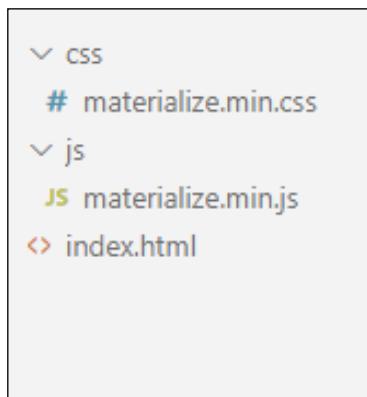


Рисунок 4

Подключаем стилевой и скриптовый файл в *index.html*:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <!-- для отображения иконок подключаем шрифты
        онлайн -->
    <link href="https://fonts.googleapis.com/
          icon?family=Material+Icons"
          rel="stylesheet">
    <!-- подключение стилевого файла фреймворка
        Materialize -->
    <link type="text/css"
          rel="stylesheet"
          href="css/materialize.min.css"
          media="screen,projection"/>
    <meta name="viewport"
          content="width=device-width,
                  initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <!-- здесь будет код страницы -->
    <!--подключение скриптового файла фреймворка
        Materialize-->

    <script type="text/javascript"
           src="js/materialize.min.js"></script>
</body>
</html>
```

Собственные стилевые и скриптовые файлы проекта нужно подключать строчкой ниже, после подключения файлов фреймворка.

Если вы не хотите скачивать Materialize в свой проект, вы можете подключить его используя [cdnjs](#) (*Content Delivery Network* — сеть компьютеров, обеспечивающая быструю доставку контента), прописав соответствующие строки:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <!-- для отображения иконок подключаем шрифты
        онлайн -->
    <link href="https://fonts.googleapis.com/
          icon?family=Material+Icons"
          rel="stylesheet">
    <link rel="stylesheet"
          href="https://cdnjs.cloudflare.com/ajax/
              libs/materialize/1.0.0/css/
              materialize.min.css">
    <meta name="viewport"
          content="width=device-width,
                  initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <!-- здесь будет код страницы -->
    <script src="https://cdnjs.cloudflare.com/ajax/
              libs/materialize/1.0.0/js/
              materialize.min.js"></script>
</body>
</html>
```

4. Цветовая палитра в Materialize

Для установки цвета содержимого или фона элементу Materialize предлагает использовать палитру цветов, которая базируется на основных цветах с использованием оттенков.

Для установки цвета фона, необходимо прописать основное имя класса, которое совпадает с названием цвета — так мы получим основной цвет из палитры Materialize.

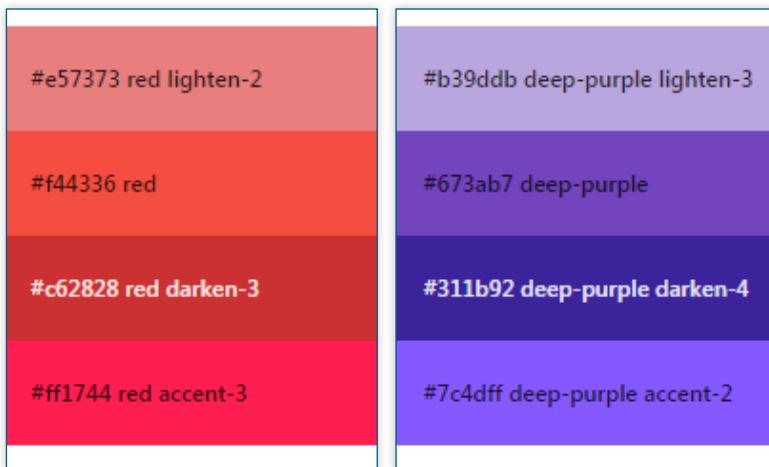


Рисунок 5

Пример палитры цветов

Для получения оттенков, необходимо прописать добавочный класс:

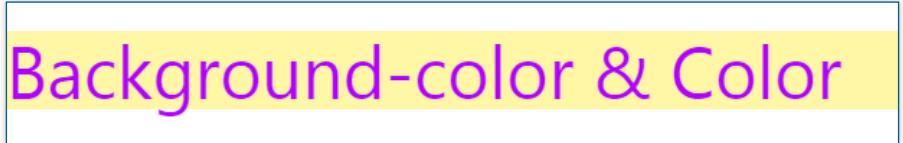
- Для более светлых — **lighten-n**, где **n** — число от **1** до **5** включительно, **5** соответствует самому светловому оттенку;
- Для более темных — **darken-n**, где **n** — число от **1** до **4** включительно, **4** соответствует самому темному оттенку;
- Для более ярких — **accent-n**, где **n** — число от **1** до **4** включительно, **4** соответствует самому яркому оттенку.

Для установки цвета текста, необходимо к названию цвета добавить «`-text`».

Например, создадим заголовок второго уровня, у которого будут светло-желтый фон и сиреневый цвет текста.

```
<h2 class="yellow lighten-3 purple-text text-accent-4">  
    Background-color & Color  
</h2>
```

В окне браузера мы видим следующее:



Background-color & Color

Рисунок 6

В следующем примере создадим кнопку, у которой будут темно-сиреневый фон и оранжевый цвет текста для акцента.

```
<button class="btn purple darken-3 orange-text  
        text-accent-3">  
    Accent button  
</button>
```

Так кнопка выглядит в окне браузера:



Рисунок 7

В следующем примере у абзаца будут оттенки серого для фона и цвета тела абзаца:

```
<p class="blue-grey lighten-5 grey-text text-darken-4">  
    Lorem ipsum dolor sit amet, consectetur  
    adipisicing elit. Placeat at voluptatem  
    laboriosam eveniet quos nisi impedit accusamus,  
    consequuntur eaque. Eum dolorem animi ratione.  
    Numquam, vel! Ab ex est ut sunt!  
</p>
```

В окне браузера абзац выглядит следующим образом:

Логотипы, изображения, таблицы и другие элементы, которые не являются текстом, не поддаются стилизации с помощью классов, описанных в этом разделе.

Рисунок 8

Просмотреть все цвета палитры можно пройдя по ссылке: materializecss.com/color.html.

5. Система сеток в Materialize

Materialize использует стандартную 12-ти колоночную сетку, которая предназначена для простого и гибкого упорядочивания элементов на странице.

Все 12 колонок имеют одинаковую ширину и располагаются горизонтально в линию в элементе-строке, при этом занимая всю ширину окна браузера. Если же вы хотите оставить отступы по бокам страницы, то следует использовать контейнер, прописав родительскому элементу класс «**container**».

Для создания адаптивного дизайна, сетка Materialize предполагает четыре размера дисплея, каждому из которых назначен соответствующий префикс класса и ширина контейнера.

Устройство, ширина	Префикс класса	Ширина контейнера
Мобильные телефоны, не более 600px	.s	90%
Планшеты, более 600px	.m	85%
Мониторы, более 992px	.l	70%
Широкоформатные мониторы, более 1200px	.xl	70%

Если вы хотите, чтобы все элементы выглядели на всех устройствах одинаково, то достаточно прописать префикс самого маленького устройства.

Все элементы, представляющие из себя «колонки», находятся в родительском элементе строке с классом «`row`».

Для того, чтобы элемент стал колонкой, нужно прописать ему класс «`col`». Для того, чтобы обозначить количество колонок, которые будет занимать элемент нужно в классах указать целое число от `1` до `12` включительно после префикса, обозначающего устройство, например, «`s12`».

Давайте рассмотрим на примерах как можно располагать элементы с помощью сетки. Для наглядности добавим фоновый цвет. Создадим строку на максимальное количество элементов — `12`.

```
<div class="container">
  <div class="row">
    <div class="red lighten-2 col s1">1</div>
    <div class="red lighten-2 col s1">2</div>
    <div class="red lighten-2 col s1">3</div>
    <div class="red lighten-2 col s1">4</div>
```

```

<div class="red lighten-2 col s1">5</div>
<div class="red lighten-2 col s1">6</div>
<div class="red lighten-2 col s1">7</div>
<div class="red lighten-2 col s1">8</div>
<div class="red lighten-2 col s1">9</div>
<div class="red lighten-2 col s1">10</div>
<div class="red lighten-2 col s1">11</div>
<div class="red lighten-2 col s1">12</div>
</div>
</div>

```

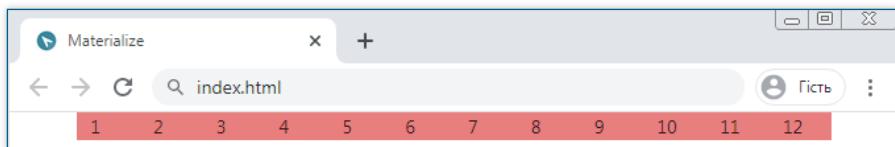


Рисунок 9

На рисунке выше видно, что контейнер с элементами расположен по середине окна браузера и все 12 элементов расположены в одну строку. Независимо от устройства, элементы будут вести себя одинаково, так как прописан префикс класса для самого маленького устройства.

Что будет, если мы изменим одному из элементов количество занимаемых колонок? Например, в первый элемент пропишем следующий класс «**blue lighten-2 col s6**». В этом классе мы прописываем цвет светло-синий цвет фона, а также класс колонки элемента и количество занимаемых колонок для всех устройств равное 6.

На рисунке 10 видно, что элементы, которым «не хватило места» переместились на следующую строку. Так произошло, потому что в строке не должно быть в сумме более 12 колонок.

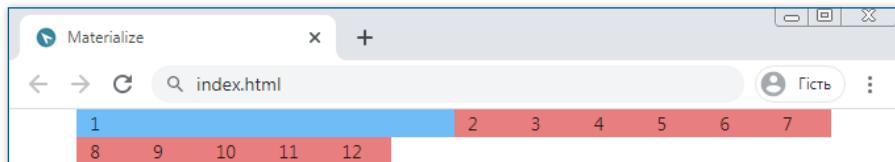


Рисунок 10

Рассмотрим, как использовать сетку для адаптивного дизайна. Создадим строку из четырех элементов, пропишем классы таким образом, чтобы на мобильных устройствах элементы занимали всю ширину контейнера, на планшетах становились по два в ряд, на мониторах все четыре в ряд и на широкоформатных мониторах элементы, расположенные посередине будут шире боковых. Для наглядности все элементы будут разного цвета.

```
<div class="container">
  <div class="row">
    <div class="red col s12 m6 l3 xl2">1</div>
    <div class="lime col s12 m6 l3 xl4">2</div>
    <div class="teal col s12 m6 l3 xl4">3</div>
    <div class="pink col s12 m6 l3 xl2">4</div>
  </div>
</div>
```

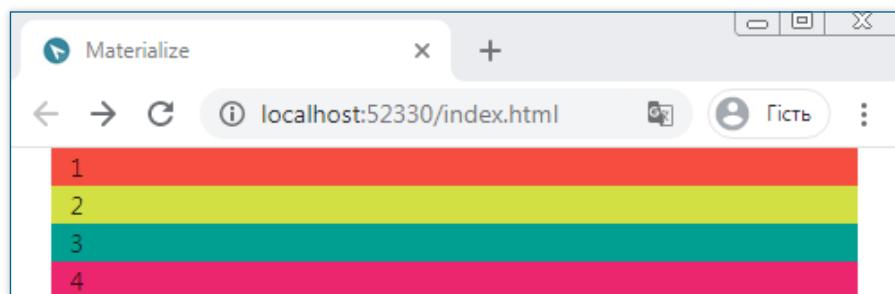


Рисунок 11. Отображение на мобильном экране

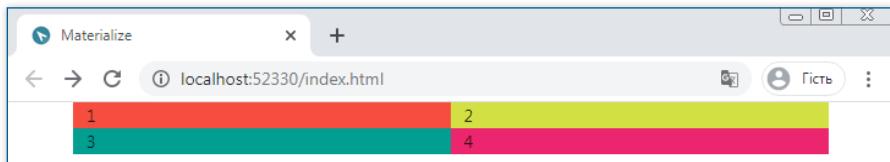


Рисунок 12. Отображение на планшете

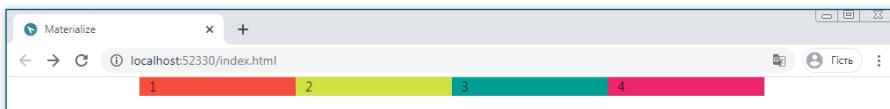
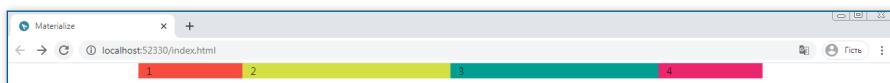


Рисунок 13. Отображение на мониторе

Рисунок 14. Отображение
на широкоформатном мониторе

Также сетка позволяет делать «разрывы» между элементами. Для этого нужно элементу, перед которым необходимо сделать отступ, нужно прописать класс **offset** с префиксом. Рассмотрим применение описанного класса на примере — добавим три элемента, два из которых будут располагаться по краям страницы, а третий будет находиться посередине один в ряду.

Для этого добавим следующий код в *index.html* в конец тега с классом «**row**»:

```
<div class="purple col s5">5</div>
<div class="blue col s5 offset-s2">6</div>
<div class="orange col s6 offset-s3">7</div>
```

Теперь наша страница на мобильном выглядит так:

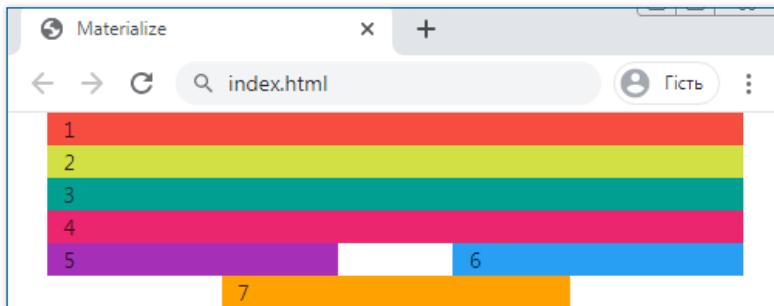


Рисунок 15

Последние три элемента будут одинаково отображаться на всех устройствах, так как класс задан для самого маленького устройства.

6. Понятие Helper. Виды Helpers

Фреймворк Materialize предлагает различного рода **helpers** — это классы, которые помогают быстро и легко отформатировать отображаемый контент.

Все **helpers** можно разделить на 4 вида:

- Выравнивание;
- Скрытие/отображение;
- Форматирование;
- Параметры браузера.

Рассмотрим каждый из видов более подробно.

Выравнивание

Для выравнивания по горизонтали предназначено три класса:

- **left-align** — по левому краю;
- **right-align** — по правому краю;
- **center-align** — по центру.

Для выравнивания по вертикали по центру предна-
значен класс `v-align-wrapper`.

Скрытие/отображение

Для скрытия элемента используются классы:

Класс	Значение
<code>hide</code>	Скрыть на всех устройствах
<code>hide-on-small-only</code>	Скрыть только на мобильных
<code>hide-on-med-only</code>	Скрыть только на планшетах
<code>hide-on-med-and-down</code>	Скрыть на планшетах и меньших устройствах
<code>hide-on-med-and-up</code>	Скрыть на планшетах и больших устройствах
<code>hide-on-large-only</code>	Скрыть только для мониторов
<code>show-on-small</code>	Показать только на мобильных
<code>show-on-medium</code>	Показать только на планшетах
<code>show-on-large</code>	Показать только на мониторах
<code>show-on-medium-and-up</code>	Показать на планшетах и меньших устройствах
<code>show-on-medium-and-down</code>	Показать на планшетах и больших устройствах

Форматирование

К группе `helpers` для форматирования относятся сле-
дующие классы:

- Сокращение «`truncate`» — для обрезания длинного текста, который выходит за рамки блока.
- Наведение «`hoverable`» — для добавления тени к элементу, на который наводится курсор.

Параметры браузера

К группе `helpers`, управляющей параметрами браузе-
ра, относится класс «`browser-default`», которые позволяет

восстановить стили следующих элементов, задаваемые браузером по умолчанию:

- тег «`ul`» — восстанавливает маркеры;
- тег «`select`» — восстанавливает отображение тега «`select`», задаваемое браузером по умолчанию;
- тега «`input`» — восстанавливает отображение тега «`input`», задаваемое браузером по умолчанию.

Рассмотрим на практике применение helpers. Создадим блочный элемент-строку с двумя элементами: заголовком и абзацем. Тег строки будет с фоном серо-голубого цвета и выравниваем контента по вертикали, а также добавлением тени при наведении. Для того, чтобы выполнить вертикальное выравнивание, зададим дополнительно высоту для строки. Заголовок будет занимать все 12 колонок с выравниванием контента по середине по горизонтали и отображаться только на мобильных отображениях. Абзац будет прятаться на мобильных отображениях и показываться на всех остальных отображениях, а также у него будет обрезаться текст.

```
<div class="row blue-grey valign-wrapper hoverable" style="height: 150px;">
    <h5 class="col s12 center-align show-on-small hide-on-med-and-up">
        center vertically aligned
    </h5>
    <p class="col s6 offset-s3 truncate hide-on-small-only">
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Veritatis illo assumenda quidem sint distinctio, reprehenderit nihil
    </p>
</div>
```

```
maiores earum voluptatibus voluptate cum  
libero impedit sit eos rerum dolorum  
reiciendis aut vitae.  
</p>  
</div>
```

На мобильном устройстве строка выглядит следующим образом:

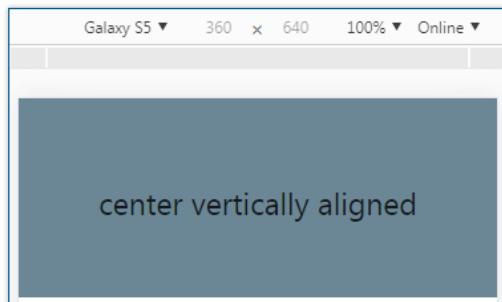


Рисунок 16

На планшетах и более широких мониторах строка выглядит как на изображении ниже (рис. 17).

Из примера видно, что для достижения приведенных эффектов достаточно добавить нужные имена классов соответствующим элементам.

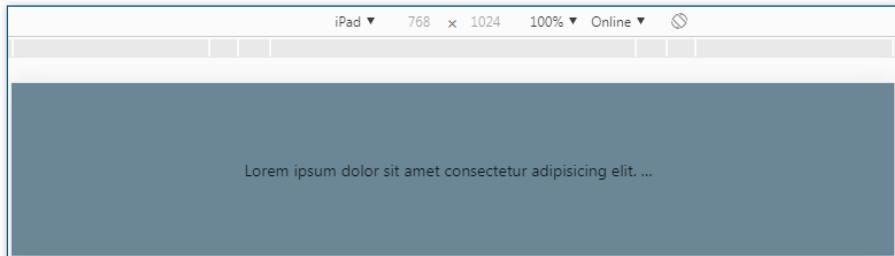


Рисунок 17

7. Навигация и Materialize

В процессе создания веб-страниц очень часто приходится создавать однотипные элементы, например, добавлять навигацию по сайту, которая фактически отличается количеством ссылок. Для решения подобных задач фреймворк Materialize предлагает различные компоненты: кнопки, карточки, пагинация, элемент предварительной загрузки. Примеры нескольких компонент:



BUTTON

Рисунок 18. Элемент с классом «btn»

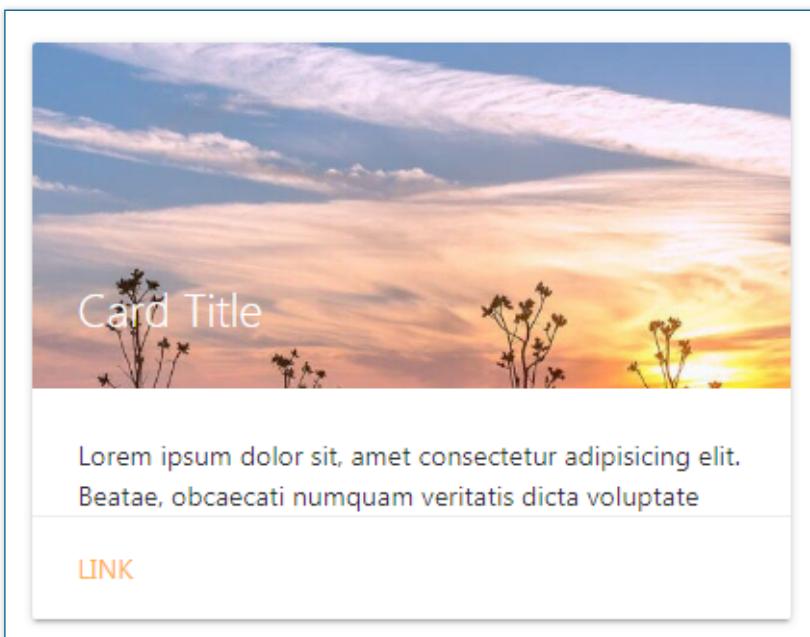


Рисунок 19. Карточка с изображением



Рисунок 20

Пагинация

Для использования любого из компонентов при верстке, необходимо соответствующему элементу добавить класс. Например, для отображения ссылки в виде кнопки, достаточно элементу добавить класс «`btn`». Для отображения блочного элемента в виде карточки, следует использовать класс «`card`». Для реализации пагинации необходимо списку, содержащему ссылки, добавить класс «`pagination`». Эти и другие компоненты описаны в разделе `Component` основной документации фреймворка `Materialize`.

Рассмотрим подробно компонент, который добавляет навигацию на страницу. Создадим меню из четырех пунктов, которое будет скрываться на планшетном и мобильном отображении.

```
<!-- добавляем элемент навигации с серо-голубым фоном -->
<nav class="blue-grey">

<!-- добавляем контейнер для основного меню,
которое будет отображаться на всех представлениях,
кроме мобильного и планшетного-->
<div class="nav-wrapper">

<!-- добавляем логотип -->
<a href="#" class="brand-logo">Logo</a>
<!-- добавляем элемент с иконкой,
по нажатию на которую, слева будет
```

```
выезжать меню на планшетных и мобильных
отображениях -->

<!-- для осуществления связи между
ссылкой и скрываемым меню, необходимо
добавить ссылке атрибут data-target -->
<a href="#" 
    data-target="mobile-left"
    class="sidenav-trigger">
    <i class="material-icons">menu</i>
</a>

<!--меню со страницами, с выравниванием
по правому краю-->
<ul class="right hide-on-med-and-down">
    <li><a href="">Page 1</a></li>
    <li><a href="">Page 2</a></li>
    <li><a href="">Page 3</a></li>
    <li><a href="">Page 4</a></li>
</ul>
</div>
</nav>

<!-- меню со страницами, которое будет выезжать
слева на мобильном и планшетном представлении -->

<!-- для осуществления связи между ссылкой
и скрываемым меню, необходимо добавить меню
идентификатор с таким же значением, как
у атрибута data-target у ссылки-->
<ul class="sidenav" id="mobile-left">
    <li><a href="">Page 1</a></li>
    <li><a href="">Page 2</a></li>
    <li><a href="">Page 3</a></li>
    <li><a href="">Page 4</a></li>
</ul>
```

В окне браузера навигация выглядит так:



Рисунок 21. На мониторах и больших устройствах

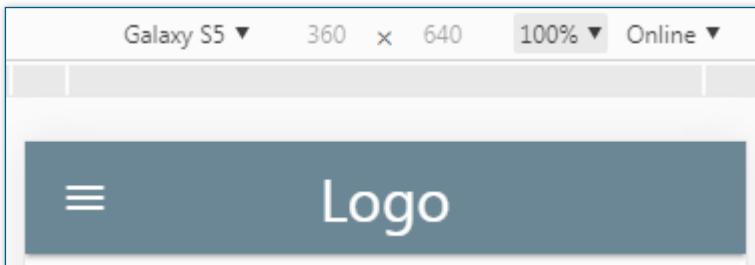


Рисунок 22. На мобильных и планшетах
в свернутом виде

Пока что при клике на иконку меню не выезжает. Для того, чтобы исправить этот момент, необходимо добавить следующий скрипт сразу после тега, который подключает скриптовый файл **materialize**.

```
<script>
    //дожидаемся загрузки страницы
    document.addEventListener('DOMContentLoaded',
        function() {
            //переменная для элемента с классом sidenav
            var elems = document.querySelectorAll('.sidenav');
            //инициализируем плагин Sidenav для
            //мобильного меню
            var instances = M.Sidenav.init(elems);
        });
</script>
```

Теперь при клике на иконку меню выезжает слева. Так это выглядит в браузере:

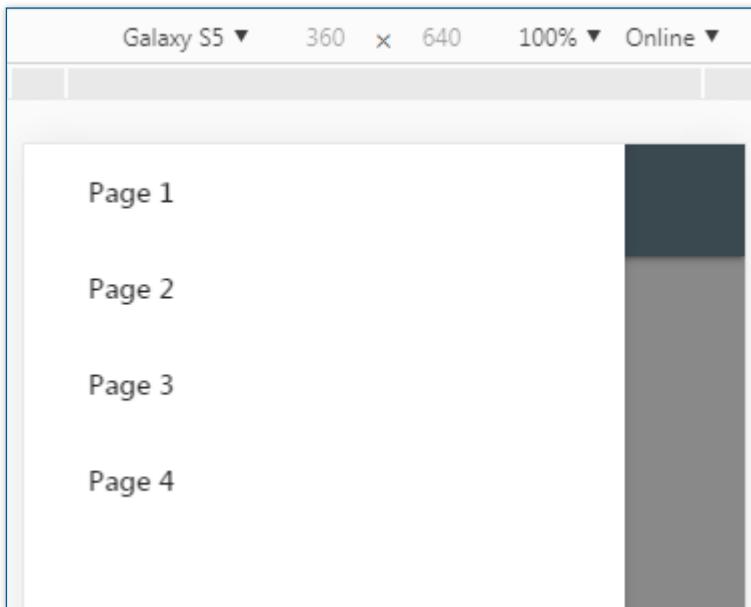


Рисунок 23. На мобильных и планшетах
в развернутом виде

8. Использование изображений и видео

Для простоты стилизации изображений фреймворк Materialize предлагает использовать следующие классы:

- **responsive-img** — для адаптивности изображения. При добавлении этого класса изображению, у последнего устанавливаются свойства «`max-width: 100%`» и «`height:auto`». При установке этих свойств максимальная ширина изображения будет равна ширине родительского элемента, а высота будет изменяться пропорционально реальной высоте изображения.

Если же картинка меньше родительского элемента, то она не изменит свои реальные размеры.

- **circle** — для того, чтобы изображения отображались круглыми.

Рассмотрим использование описанных классов на примере. Создадим строку, в которой будет три колонки для трех изображений.

```
<div class="container">
    <div class="row">
        <div class="col s4">
            <!-- первое изображение без класса -->
            
        </div>

        <div class="col s4">
            <!-- второе изображение с классом
            responsive-img -->
            
        </div>

        <div class="col s4">
            <!-- третье изображение с классами
            responsive-img и circle -->
            
        </div>
    </div>
</div>
```

В браузере видно, что первое изображение отображается в своих реальных размерах, а остальные два имеют ширину равную 100% ширины родительского элемента.

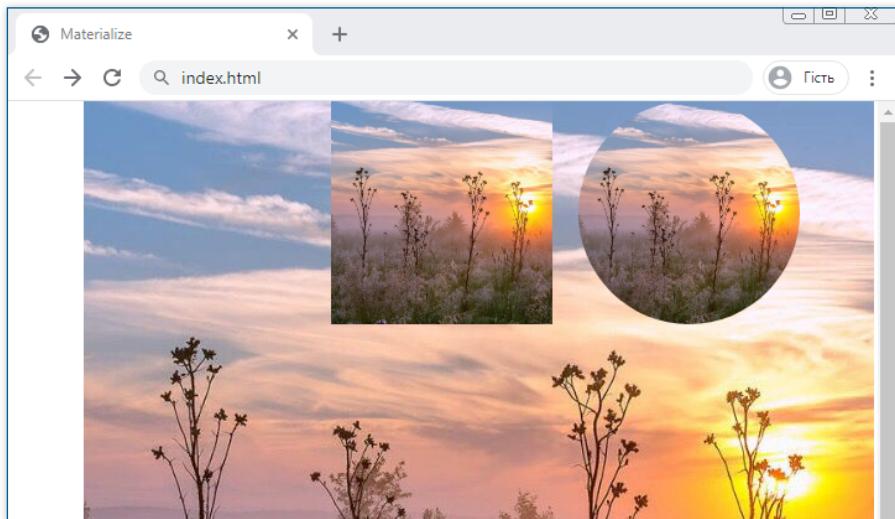


Рисунок 24

Применение классов для изображений

Также существуют классы для адаптивности встраиваемого видео. Для этого надо добавить класс «`video-container`» для родительского элемента, который содержит тег `iframe`, либо добавить класс «`responsive-video`» в тег `video`.

Рассмотрим на примере использование описанных классов. Создадим строку на две колонки. В первой колонке с помощью тега `<iframe>` подключим видео с youtube, а во второй колонке подключим локальный видео файл с помощью тега `<video>`.

```
<div class="row">
    <div class="col s6">
        <!-- добавляем класс video-container блочному
            элементу-контейнеру для адаптивности -->
```

```
<div class="video-container">
    <iframe src="https://www.youtube.com/
        embed/rrT6v5s0wJg"
        frameborder="0"
        allowfullscreen>
    </iframe>
</div>
<h6 class="center">Making Material Design</h6>
</div>
<div class="col s6">
    <!-- добавляем класс responsive-video для
    адаптивности -->
    <video class="responsive-video" controls>
        <source src="media/palette.mp4"
            type="video/mp4">
    </video>
    <h6 class="center">Palette Perfect</h6>
</div>
</div>
```

В окне браузера видно, что видео занимает всю ширину родительского элемента.

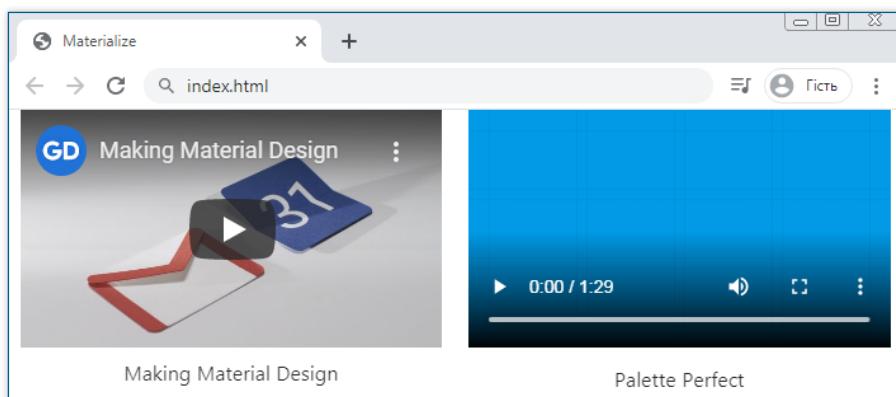


Рисунок 25

9. Скроллинг

Скроллинг — это плавная прокрутка к определенному месту на странице. Например, при создании одностраницочного сайта, каждый пункт меню ведет к определенному разделу на странице. Если не применять скроллинг, то переход будет мгновенный и пользователь может его не заметить.

Элементу, к которому нужно выполнить плавную прокрутку, добавляют идентификатор и класс «`scrollspy`».

Рассмотрим применение скроллинга на примере. Создадим строку, в которой будет два элемента на 9 и 3 колонки. Больший элемент будет содержать контент, меньший — меню с якорями.

```
<div class="row">
    <div class="col s9">
        <div id="section1" class="section scrollspy">
            <h3>Section 1</h3>
            <p>Content</p>
        </div>

        <div id="section2" class="section scrollspy">
            <h3>Section 2</h3>
            <p>Content</p>
        </div>

        <div id="section3" class="section scrollspy">
            <h3>Section 3</h3>
            <p>Content</p>
        </div>
    </div>

    <div class="col s3">
```

```
<!-- добавим фиксированное позиционирование,  
чтобы элемент с ссылками всегда находился  
в определенном месте -->  
<ul class="section table-of-contents"  
    style="position: fixed;">  
    <li><a href="#section1">section 1</a></li>  
    <li><a href="#section2">section 2</a></li>  
    <li><a href="#section3">section 3</a></li>  
</ul>  
</div>  
</div>
```

Для более наглядной демонстрации вместо слова «Content» в абзаце, следует вставить текст-рыбу. В браузере страница выглядит так:

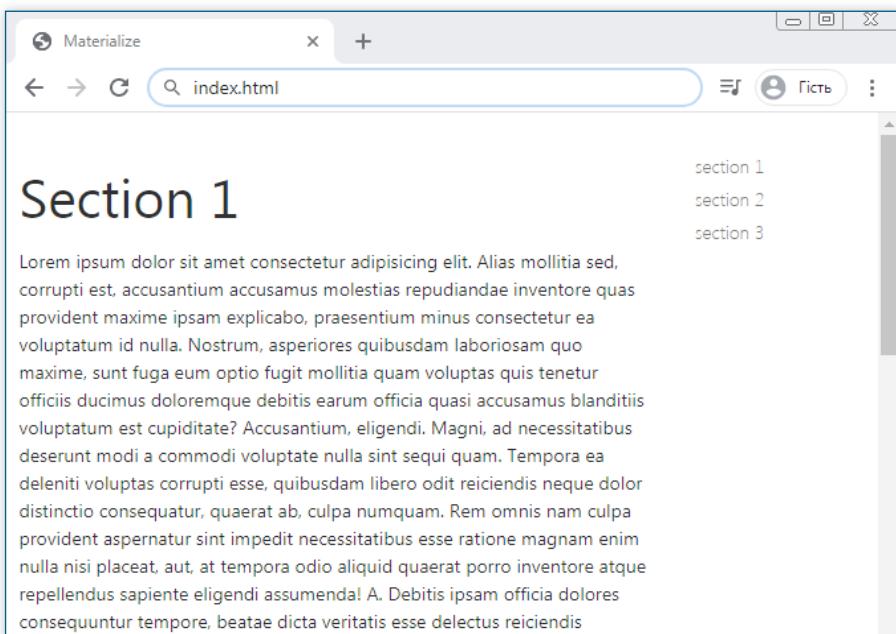


Рисунок 26

Переход к секции пока что осуществляется мгновенно без плавной прокрутки. Для того, чтобы это исправить, необходимо добавить следующий скрипт, сразу же после скрипта, подключающего файл **Materialize**.

```
<script>
    document.addEventListener('DOMContentLoaded',
        function() {
            //переменная для элемента с классом scrollspy
            var elems = document.querySelectorAll('.scrollspy');
            //настройки опций для скролла
            var options = {
                //дроссель обработчика прокрутки
                throttle:500,
                //смещение до прокручиваемого элемента от
                //верхнего края окна браузера
                scrollOffset:5,
                //класс для ссылки, которую выбрали
                activeClass:'active',
                //функция, которая определяет активный
                // элемент при скролле страницы
                getActiveElement: function(id) {
                    return 'a[href="#' + id + '"]';
                }
            }

            //инициализация плагина ScrollSpy
            var instances = M.ScrollSpy.init(elems,options);
        });
</script>
```

Теперь при клике на ссылку справа страница плавно прокручивается к нужной секции. При желании, можно переопределить настройки прокрутки. В браузере страница выглядит так:

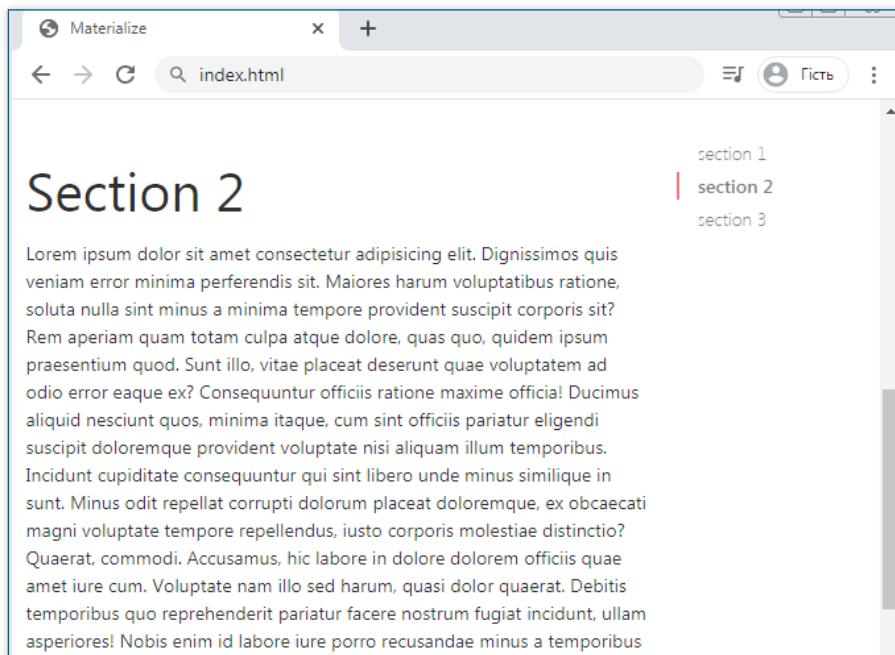


Рисунок 27

На рисунке видно, что активная ссылка выделена красной рамкой слева и более жирным начертанием текста.

10. Пример создания сайта с помощью фреймворка Materialize

Рассмотрим на практике создание небольшого сайта с использованием фреймворка Materialize.

В проекте будет 3 страницы: домашняя, блог и контакты. Две из которых, блог и контакты, вам нужно будет завершить в качестве домашнего задания.

Домашняя страница будет состоять из трех секций. Первая секция будет содержать заголовок и форму обратной связи, во второй секции будет три текстовых блока,

и в третьей секции будут отображаться карточки с фотографиями команды.

Навигационное меню и подвал будут одинаковыми для всех страниц. Навигационное меню будет фиксироваться по верхнему краю окна браузера.

На рисунках ниже показано как будут выглядеть описанные секции:

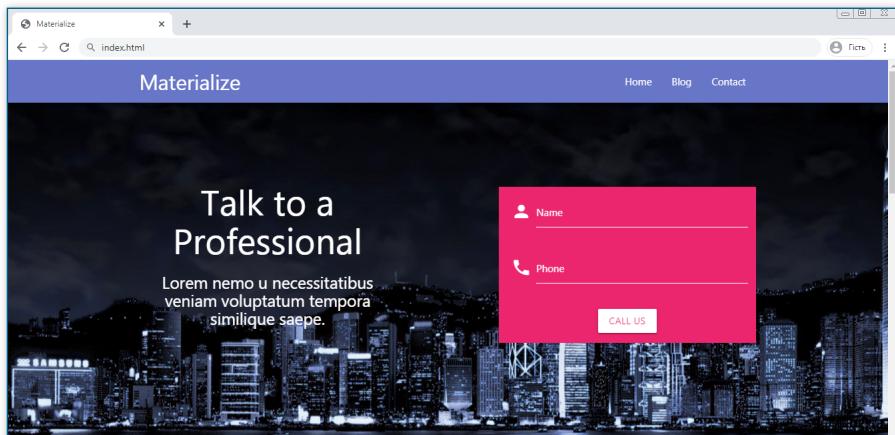


Рисунок 28. Первая секция

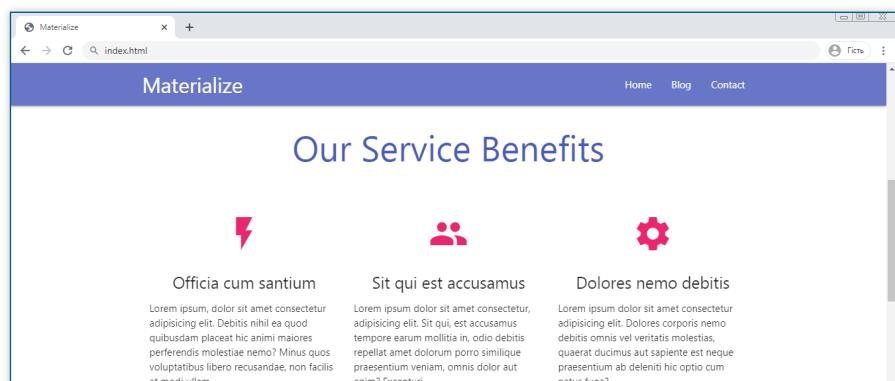


Рисунок 29. Вторая секция

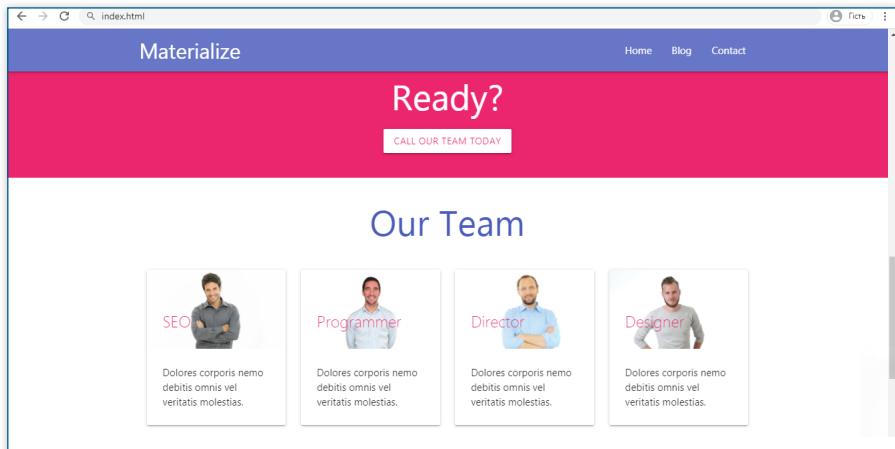


Рисунок 30. Третья секция

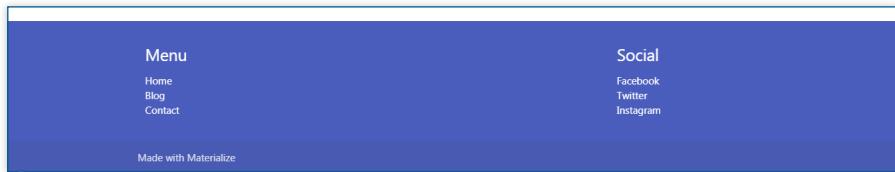


Рисунок 31. Подвал

Создадим проект, который будет иметь следующую структуру:

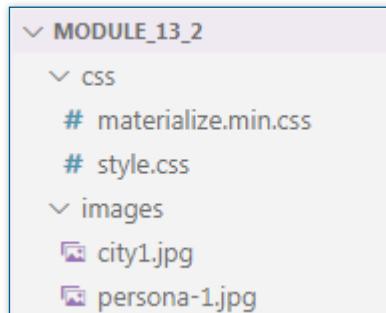


Рисунок 32. (Начало)

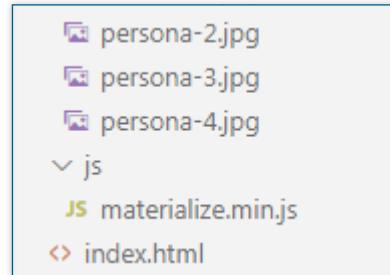


Рисунок 32. (Продолжение)

Основной код будет содержать страница *index.html*. Для начала подключим стилевой и скриптовый файл фреймворка.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <link href="https://fonts.googleapis.com/
          icon?family=Material+Icons"
          rel="stylesheet">
    <link rel="stylesheet"
          href="css/materialize.min.css">
    <link rel="stylesheet" href="css/style.css">
    <meta name="viewport"
          content="width=device-width,
                  initial-scale=1.0">
    <title>Materialize</title>
</head>

<body>
    <script src="js/materialize.min.js"></script>
</body>

</html>
```

Приступим к созданию навигации, воспользуемся компонентом navbar (materializecss.com/navbar.html):

```
<!-- для фиксирования навигации по верхнему краю
создадим блочный контейнер с классом navbar-fixed -->
<div class="navbar-fixed">
    <nav class="indigo lighten-1" role="navigation">
        <div class="nav-wrapper container">
            <a id="logo-container"
                href="#"
                class="brand-logo"> Materialize
            </a>
            <!--основное меню прячется на планшетном
            отображении и меньше-->
            <ul class="right hide-on-med-and-down">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="blog.html">Blog</a></li>
                <li><a href="contact.html">Contact</a></li>
            </ul>
            <!-- для создания выезжающего меню создадим
            список с классом sidenav -->
            <ul id="nav-mobile" class="sidenav">
                <li class="active"><a href="#">Home</a></li>
                <li><a href="blog.html">Blog</a></li>
                <li><a href="contact.html">Contact</a></li>
            </ul>
            <!--иконка для мобильного меню -->
            <a href="#" data-target="nav-mobile"
                class="sidenav-trigger">
                <i class="material-icons">menu</i>
            </a>
        </div>
    </nav>
</div>
```

Не забудьте добавить скрипт для того, чтобы меню выезжало слева в мобильном отображении.

```
<script>
    document.addEventListener('DOMContentLoaded',
                           function() {
        var elems = document.querySelectorAll('.sidenav');
        var instances = M.Sidenav.init(elems);
    });
</script>
```

Создадим первую секцию с формой обратной связи и эффектом parallax для изображения фона (materializecss.com/parallax.html). Эффектом **parallax** называется техника, когда фоновое изображение в перспективе движется медленнее, чем элементы переднего плана.

```
<!-- первая секция содержит фоновую картинку,
у которой будет эффект parallax -->
<section id="index-banner"
          class="parallax-container">
    <div class="section no-pad-bot">
        <div class="container">
            <div class="row">

                <!-- колонка для заголовка и слогана,
                используются различные классы для
                адаптивного отображения -->
                <div class="col s12 m10 offset-m1 l16 xl5">
                    <h2 class="header center white-text">
                        Talk to a Professional
                    </h2>
                    <h5 class="center white-text">
                        Lorem nemo u necessitatibus veniam
                        voluptatum tempora similique saepe.
                    </h5>
                </div>
            </div>
        </div>
    </div>
</section>
```

```
<!-- колонка для формы обратной связи,
используются различные классы для
адаптивного отображения -->
<div class="col s12 m10 offset-m1 l6
x15 offset-x12 center">
<br><br>
<div class="row pink white-text">
<form action="">
<div class="input-field col s12">
<!-- иконка пользователя -->
<i class="material-icons prefix">
person</i>
<input type="text"
id="name"
class="autocomplete
my-input">
<label for="name">Name</label>
</div>
<div class="input-field col s12">
<!-- иконка телефона -->
<i class="material-icons prefix">
phone</i>
<input type="tel"
id="tel"
class="autocomplete
my-input">
<label for="tel">Phone</label>
</div>
<div class="input-field col s12">
<button type="submit"
class="btn pink-text white">
Call us
</button>
</div>
</form>
</div>
</div>
```

```

        </div>
    </div>
</div>

<!-- подключаем фоновое изображение в контейнере
с классом parallax --&gt;
&lt;div class="parallax"&gt;
    &lt;img src="images/city1.jpg" alt="city"&gt;
&lt;/div&gt;
&lt;/section &gt;</pre>

```

Для реализации эффекта **parallax**, пропишем следующий скрипт перед закрывающим тегом **body**:

```

<script>
    document.addEventListener('DOMContentLoaded',
                           function() {
        var elems = document.querySelectorAll('.parallax');
        var instances = M.Parallax.init(elems);
    });
</script>
```

Приступим к созданию следующей секции, описывающей сервисы компании:

```

<section class="container">
    <h2 class="header center indigo-text">
        Our Service Benefits
    </h2>
    <div class="section">
        <!-- строка будет содержать три колонки
        с иконками --&gt;
        &lt;div class="row"&gt;
            &lt;div class="col s12 l4"&gt;
                &lt;div class="icon-block"&gt;</pre>

```

```
<h2 class="center pink-text">
    <i class="medium material-icons">flash_on</i>
</h2>
<h5 class="center">
    Officia cum santium
</h5>
<p>Lorem ipsum, dolor sit amet
    consectetur adipisicing elit.
    Debitis nihil ea quod quibusdam
    placeat hic animi maiores
    perferendis molestiae nemo?
    Minus quos voluptatibus libero
    recusandae, non facilis at modi
    ullam.
</p>
</div>
</div>

<div class="col s12 l4">
    <div class="icon-block">
        <h2 class="center pink-text">
            <i class="medium material-icons">group</i>
        </h2>
        <h5 class="center">
            Sit qui est accusamus
        </h5>
        <p>Lorem ipsum dolor sit amet consectetur,
            adipisicing elit. Sit qui, est
            accusamus tempore earum mollitia
            in, odio debitis repellat amet
            dolorum porro similique praesentium
            veniam, omnis dolor aut enim?
            Excepturi.
        </p>
    </div>
</div>
```

```

<div class="col s12 l4">
    <div class="icon-block">
        <h2 class="center pink-text">
            <i class="medium material-icons">
                settings</i>
        </h2>
        <h5 class="center">
            Dolores nemo debitis</h5>
        <p>Lorem ipsum dolor sit amet consectetur
            adipisicing elit. Dolores corporis
            nemo debitis omnis vel veritatis
            molestias, quaerat ducimus aut
            sapiente est neque praesentium ab
            deleniti hic optio cum natus fuga?
        </p>
    </div>
</div>
</div>
<br><br>
</section>

```

Следующая секция будет служить разделителем и содержать кнопку, которая будет переводить к форме обратной связи:

```

<section class="section no-pad-bot pink">
    <div class="container">
        <div class="row">
            <div class="col s12 center">
                <h2 class="header center white-text">
                    Ready?
                </h2>
                <p><a href="#" class="btn pink-text white">
                    Call our team today</a>
                </p>
            </div>
        </div>
    </div>
</section>

```

```
        <br>
    </div>
</div>
</div>
</section>
```

Последняя секция отображает карточки с изображениями, для реализации воспользуемся компонентом **cards** (materializecss.com/cards.html):

```
<section class="container">
    <h2 class="header center indigo-text">Our Team</h2>
    <div class="section">
        <div class="row">
            <div class="col s12 m6 xl3">
                <!-- блок для карточки -->
                <div class="card">
                    <!-- блок для изображения карточки -->
                    <div class="card-image">
                        
                    <!--надпись на изображении -->
                    <span class="card-title pink-text">
                        SEO</span>
                </div>
                <!--блок для текста карточки -->
                <div class="card-content">
                    <p>Dolores corporis nemo debitatis omnis
                       vel veritatis molestias.
                    </p>
                </div>
            </div>
        </div>
        <div class="col s12 m6 xl3">
            <div class="card">
                <div class="card-image">
```

```

<span class="card-title pink-text">
Programmer</span>
</div>
<div class="card-content">
<p>Dolores corporis nemo debitibus omnis
vel veritatis molestias.
</p>
</div>
</div>
<div class="col s12 m6 x13">
<div class="card">
<div class="card-image">

<span class="card-title pink-text">
Director</span>
</div>
<div class="card-content">
<p>Dolores corporis nemo debitibus omnis
vel veritatis molestias.
</p>
</div>
</div>
</div>
<div class="col s12 m6 x13">
<div class="card">
<div class="card-image">

<span class="card-title pink-text">
Designer</span>
</div>
<div class="card-content">
<p>Dolores corporis nemo debitibus omnis
vel veritatis molestias.
</p>
</div>
```

```
        </div>
    </div>
</div>
<br><br>
</section>
```

Подвал сайта разделен на две колонки. В первой отображается основное меню сайта, во второй — меню социальных сетей.

```
<footer class="page-footer indigo">
    <div class="container">
        <div class="row">
            <div class="col s6">
                <h5 class="white-text">Menu</h5>
                <ul>
                    <li><a class="white-text" href="#">
                        Home</a>
                    </li>
                    <li><a class="white-text"
                        href="blog.html">Blog</a>
                    </li>
                    <li><a class="white-text"
                        href="contact.html ">
                        Contact</a>
                    </li>
                </ul>
            </div>
            <div class="col s6 l3 offset-l3">
                <h5 class="white-text">Social</h5>
                <ul>
                    <li><a class="white-text"
                        href="https://facebook.com">
                        Facebook </a>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</footer>
```

```
<li><a class="white-text"  
      href="https://twitter.com">  
    Twitter </a>  
</li>  
<li><a class="white-text"  
      href="https://instagram.com">  
    Instagram</a>  
</li>  
</ul>  
</div>  
</div>  
<div class="footer-copyright">  
  <div class="container">  
    Made with Materialize  
</div>  
</div>  
</footer>
```

Домашнее задание

Добавить две страницы Блог и Контакты в проект, который создавали выше на занятии.

Протестировать отображение страниц на различных устройствах: мобильном, планшете, мониторе.

Страница Блог будет разделена на две части: первая отображает карточки со статьями (materializecss.com/cards.html), и вторая содержит списки (materializecss.com/collections.html) (рис. 33-34).

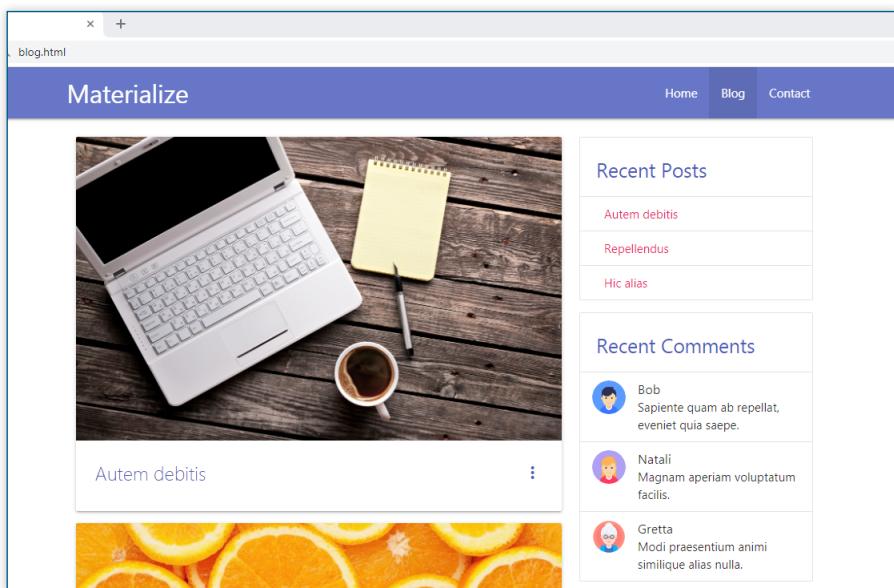


Рисунок 33

Домашнее задание

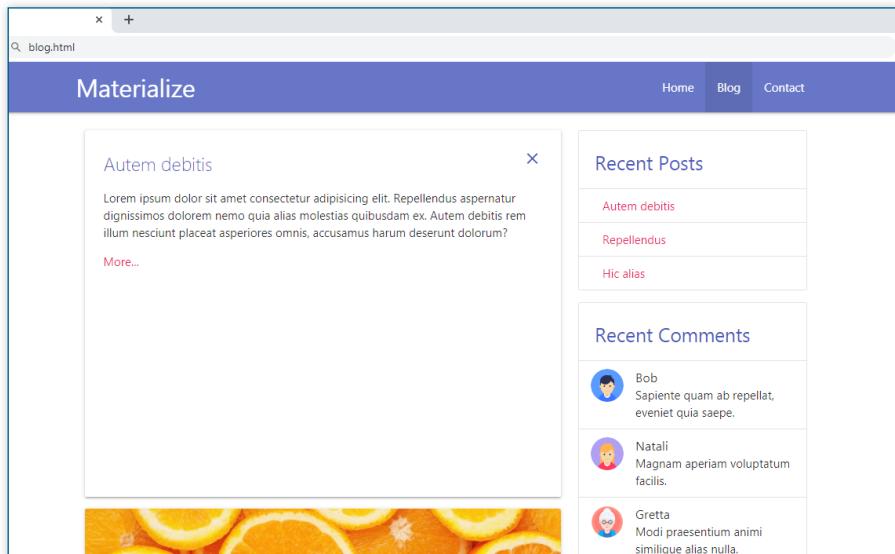


Рисунок 34

Страница Контакт будет содержать форму обратной связи (materializecss.com/text-inputs.html) (рис. 35).

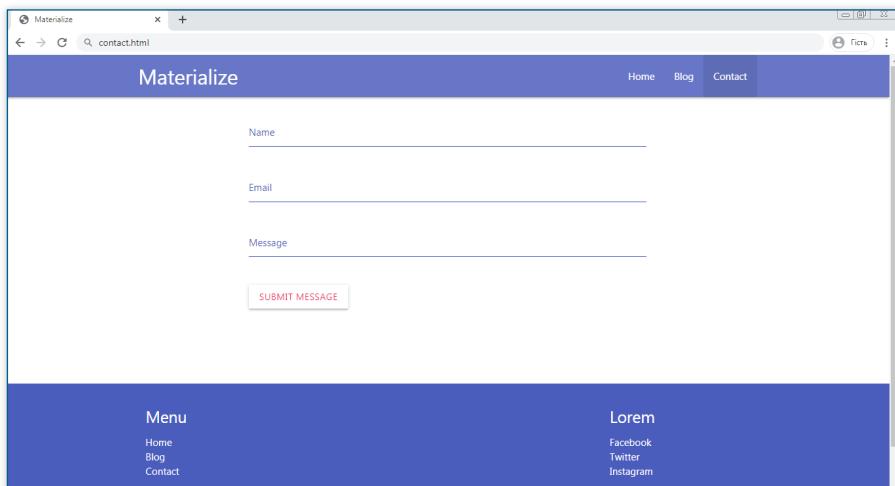


Рисунок 35



Урок 13. Materialize

© Елена Здановская.

© STEP IT Academy, www.itstep.org.

All rights to protected pictures, audio, and video belong to their authors or legal owners.

Fragments of works are used exclusively in illustration purposes to the extent justified by the purpose as part of an educational process and for educational purposes in accordance with Article 1273 Sec. 4 of the Civil Code of the Russian Federation and Articles 21 and 23 of the Law of Ukraine "On Copyright and Related Rights". The extent and method of cited works are in conformity with the standards, do not conflict with a normal exploitation of the work, and do not prejudice the legitimate interests of the authors and rightholders. Cited fragments of works can be replaced with alternative, non-protected analogs, and as such correspond the criteria of fair use.

All rights reserved. Any reproduction, in whole or in part, is prohibited. Agreement of the use of works and their fragments is carried out with the authors and other right owners. Materials from this document can be used only with resource link.

Liability for unauthorized copying and commercial use of materials is defined according to the current legislation of Ukraine.