# Реферат

Пояснительная записка курсового проекта содержит 00 страницы, 00 рисунков, 0 источников литературы, 0 приложение.

# Введение

Электронный учебник — это специальное устройство либо программное обеспечение, используемое в образовательном процессе и заменяющее собой традиционный бумажный учебник. В настоящее время трактовка словосочетания «электронный учебник» очень широка: в некоторых случаях под ним подразумевается электронная версия бумажного учебника, в некоторых — сложный комплекс программ на электронных устройствах, позволяющий демонстрировать ученикам, помимо текста, обучающий мультимедийный материал, содержащий в себе также интерактивные блоки проверки знаний, обновляющийся из централизованного источника и так далее. [1]

Практически все компании, заведения и фирмы имеют свой собственный сайт.

Сайт представляет собой множество страниц, связанных между собой общей тематической направленностью, единым оформлением (дизайном) и системой навигацией (ссылками). Сайт является информационной единицей, представляющей компанию или отдельно взятого человека в глобальной сети Интернет. А также сайт является одним из современных средств передачи информации, коммуникативным средством, и, наконец, рекламным продуктом, дающим большие возможности в области поиска и привлечения клиентов [1].

Тема курсового проекта обусловлена желанием помочь начинающим программистам, либо профессионалам узнать новое или повторить давно забытое в данной сфере.

В курсовом проекте реализован дизайн и верстка проекта «Электронное учебное издание по pythonу».

Пояснительная записка состоит из введения, 4 разделов, заключения.

Во введении описана актуальность выполняемого проекта.

В первом разделе приведен анализ предметной области.

Во втором разделе описан этап проектирования.

В третьем разделе приведено обоснование выбора стилевого решения и приведен макет.

В четвертом разделе описана реализация проекта

В пятом разделе описано тестирование и отладка.

В заключении приведены выводы о проделанной работе.

**Содержание**

[Реферат 6](#_Toc512159434)

[Введение 7](#_Toc512159435)

[1 Постановка задачи 9](#_Toc512159436)

[2 Проектирование 11](#_Toc512159437)

[3 Дизайн-проект 12](#_Toc512159438)

[4 Реализация 13](#_Toc512159439)

[5 Тестирование 14](#_Toc512159440)

[Заключение 15](#_Toc512159441)

[Список используемых источников 16](#_Toc512159442)

[Приложение 17](#_Toc512159443)

# 1. Постановка задачи

Целью данного курсового проекта было создание электронного учебного издания по python.

При выполнении данной цели проекта поставлены следующие задачи:

– Разработать интерфейс сайта в соответствии с правилами юзабилити и эргономики;

– Реализовать функционал для просмотра занятий, просмотра видеоматериалов и скачивание всей необходимой документации;

– Реализовать программно функционал веб-сайта;

– Протестировать разработанный сайт.

Целевую аудиторию можно разделить на несколько основных групп: профессионалы, любители, новички.

Профессионалы – это люди, которые хорошо разбираются в данной сфере, долго этим увлекаются, могут преподавать. Любители – люди, занимающиеся веб-дизайном для себя, то есть, как хобби. Новички – люди, которые ничего не знают об этих направлениях, но хотят начать заниматься.

Целевая аудитория данного сайта – преимущественно мужчины и женщины 16-35 лет из русскоговорящих стран. Уровень дохода – неважен. Часть из которых увлекается компьютерами с детства, разбираются в устройстве сайтов.

Персонажи:

1. Арина Янцевич (новичок)

Возраст: 18 лет.

Семейное положение: не замужем.

Профессия: абитуриентка.

Собирается поступать на программиста, хочет заранее изучить интересующий ее язык программирования.

Цель посещения:

Изучить язык программирования «python».

Потребности:

1. Новые знания.

Требования к сайту:

1. Простота поиска нужной информации.

2. Понятные занятия.

3. Возможность задать вопрос по уроку.

1. Ярослав Брановицкий (профессионал)

Возраст: 32 года.

Семейное положение: холост.

Профессия: программист.

Сотрудник успешной фирмы. Ищет учебное издание для повышения квалификации.

Цель посещения:

Повысить квалификацию.

Потребности:

1. Найти достаточно информации.

Требования к сайту:

1. Простота поиска нужной информации.

2. Качество занятий.

1. Дмитрий Самосюк(любитель)

Возраст: 20 лет.

Семейное положение: холост.

Профессия: студент.

Дмитрий студент 3 курса, ему предложили работу, но для этого ему необходимо подтянуть знания по языку программирования «python».

Цель посещения:

Подтянуть знания по языку программирования «python».

Потребности:

1. Новые умения.

2. Провести врем с удовольствием.

Требования к сайту:

1. Простота поиска нужной информации.

Аналоги:

1. https://pythonworld.ru/

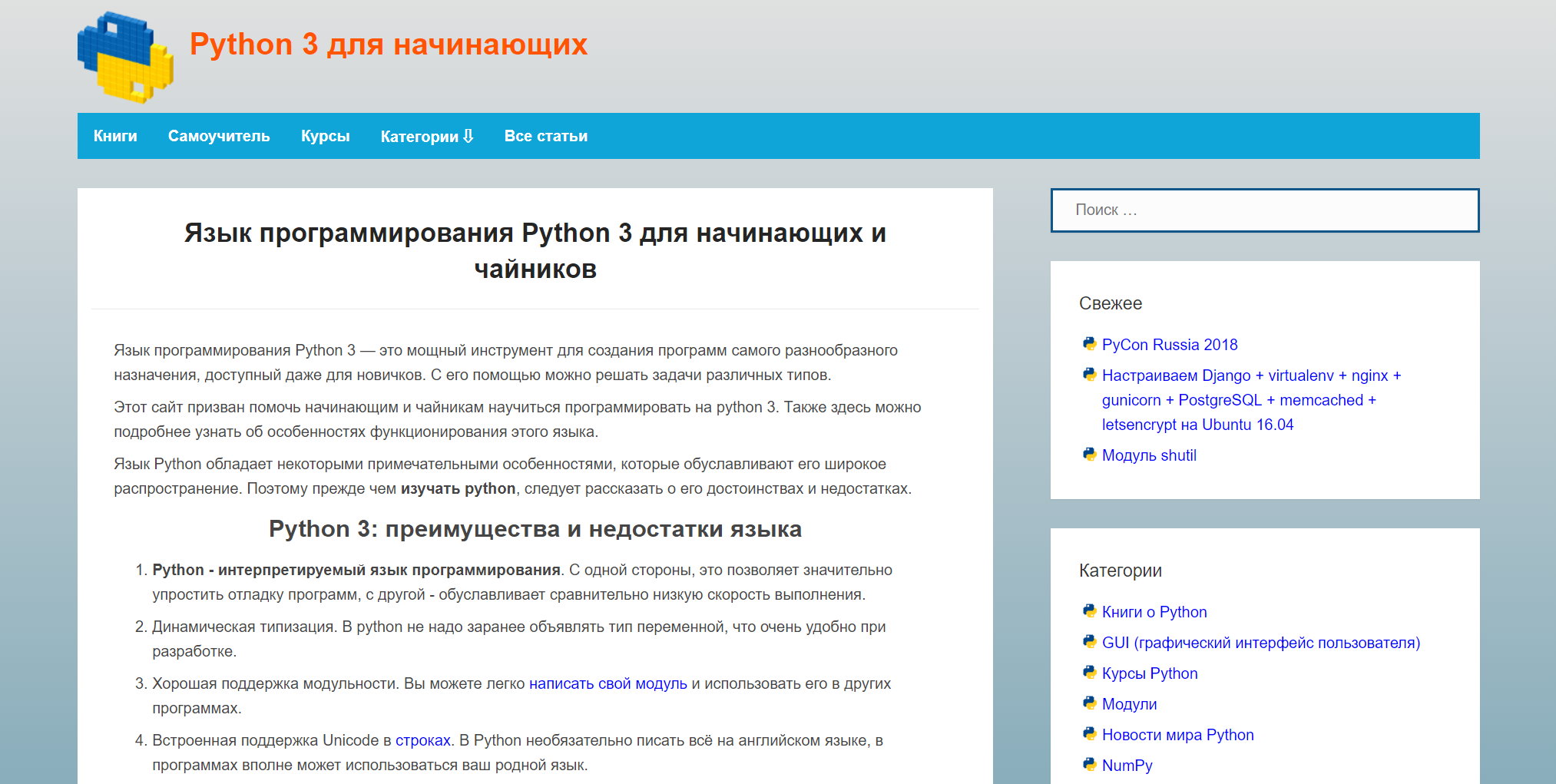


Рисунок 1.1 – Пример сайта

Самоучитель по языку python. Имеет простой, не отвлекающий дизайн. Информация разделена по блокам. Цветовая схема светлая. Дизайн сайта выдержан в единой стилистике. Четко выделяются 2 основных цвета – синий и белый. Логотип хорошо отражает то, с чем связана деятельность сайта. Логотип комбинированный – присутствуют изображение и текст.

Удобная область навигации в шапке. В главном меню выделены основные необходимые пункты. Навигация удобная. Сайт адаптивный.

1. http://pythontutor.ru/

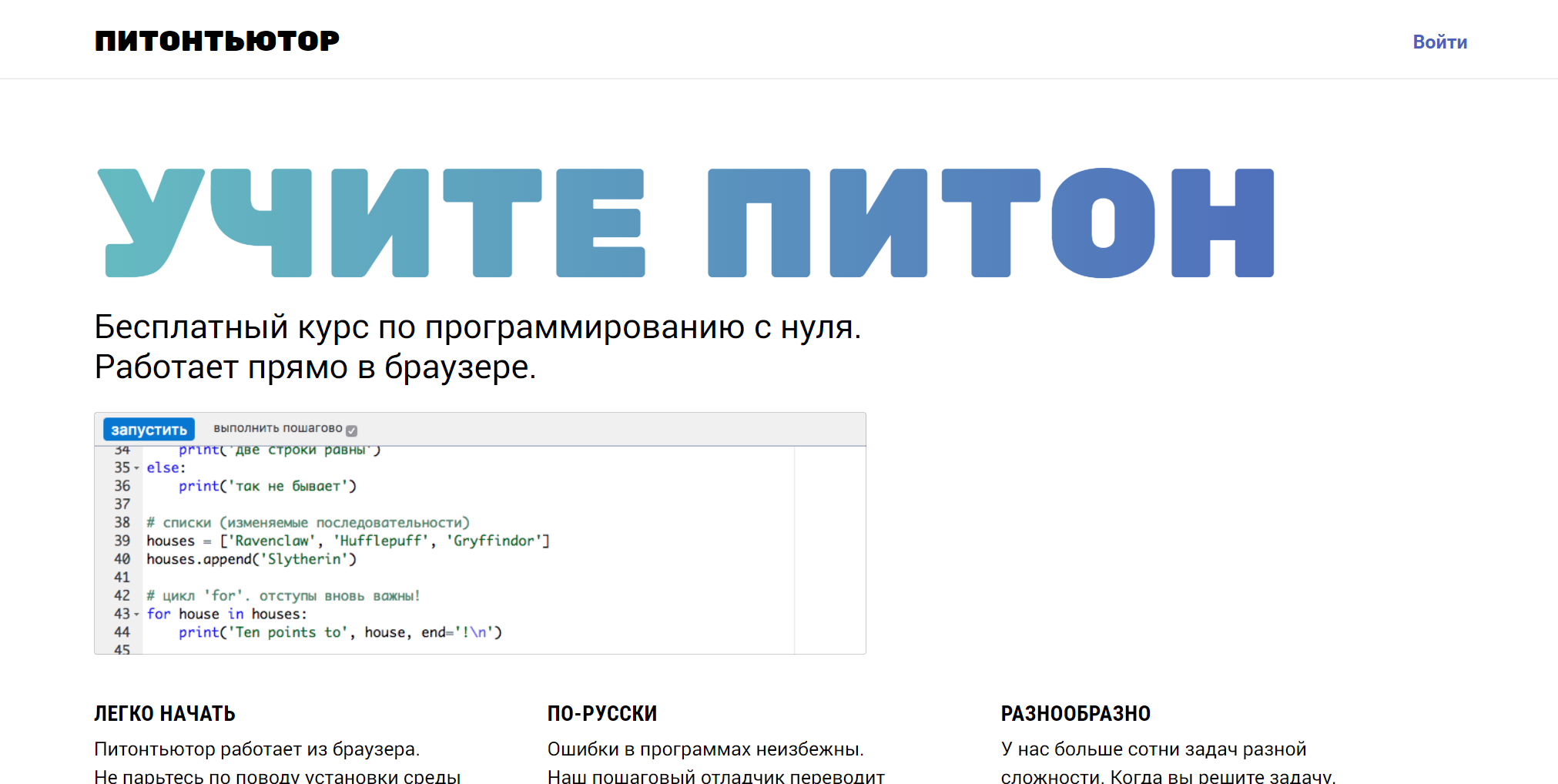


Рисунок 1.2 – Пример сайта

Еще один сайт по pythonу. Дизайн сайта выдержан в определенной стилистике. Цветовая гамма сайта светлая. Дизайн нейтральный. Сайт не адаптивный.

3. https://learnpythonthehardway.org/book/

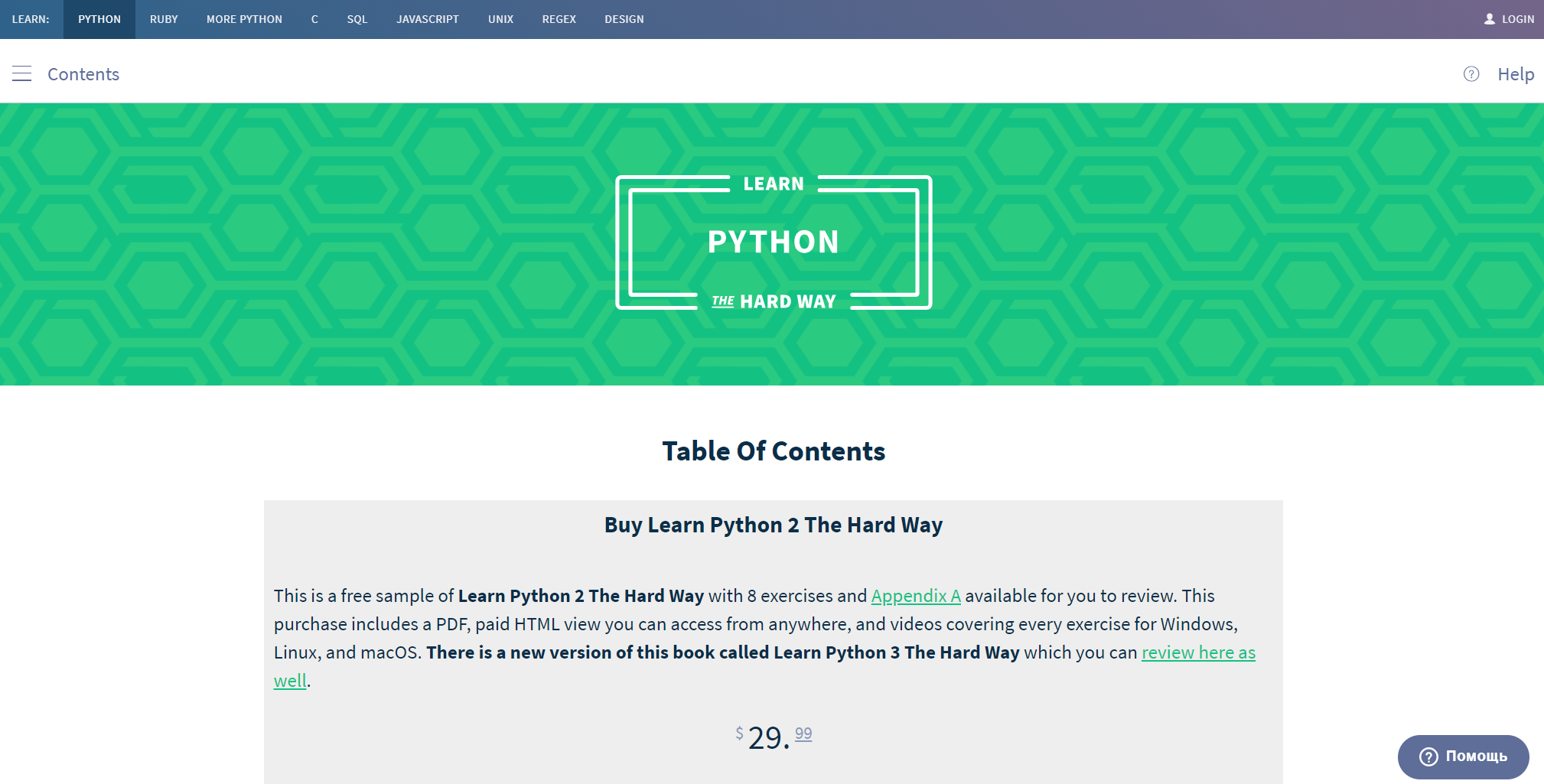


Рисунок 1.3 – Пример сайта

Сайт имеет удобное меню, а также наличие других языков для изучения. Дизайн сайта выполнен в единой стилистике. Нет лишнего. Цветовая гамма светлая. Большинство графического материала хорошего качества.

Сайт не перегружен. Выделяется одна колонка с основным содержимым. Текст хорошо читается на контрастном фоне. Сайт адаптивный. При первом входе не составляет труда найти нужную информацию. Обучение пользователя не составляет труда. Он легко восстановит путь до нужной информации при повторном посещении.

# 

# 2. Проектирование

В проектировании использовались следующие технологии:

HTML — стандартизированный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. [2]

CSS — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL. [3]

Javascript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. [4]

Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Bootstrap использует современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров. [5]

# 3. Дизайн-проект

Для создания сайта студии был выбран стиль минимализм. Основными цветами были выбраны белый и бежевый. На сайте не должно быть ничего лишнего, пользователь должен концентрироваться на обучении.

В качестве логотипа было решено использовать одну из интерпретаций логотипа «Python».

На рисунках представлены макеты страниц сайта:

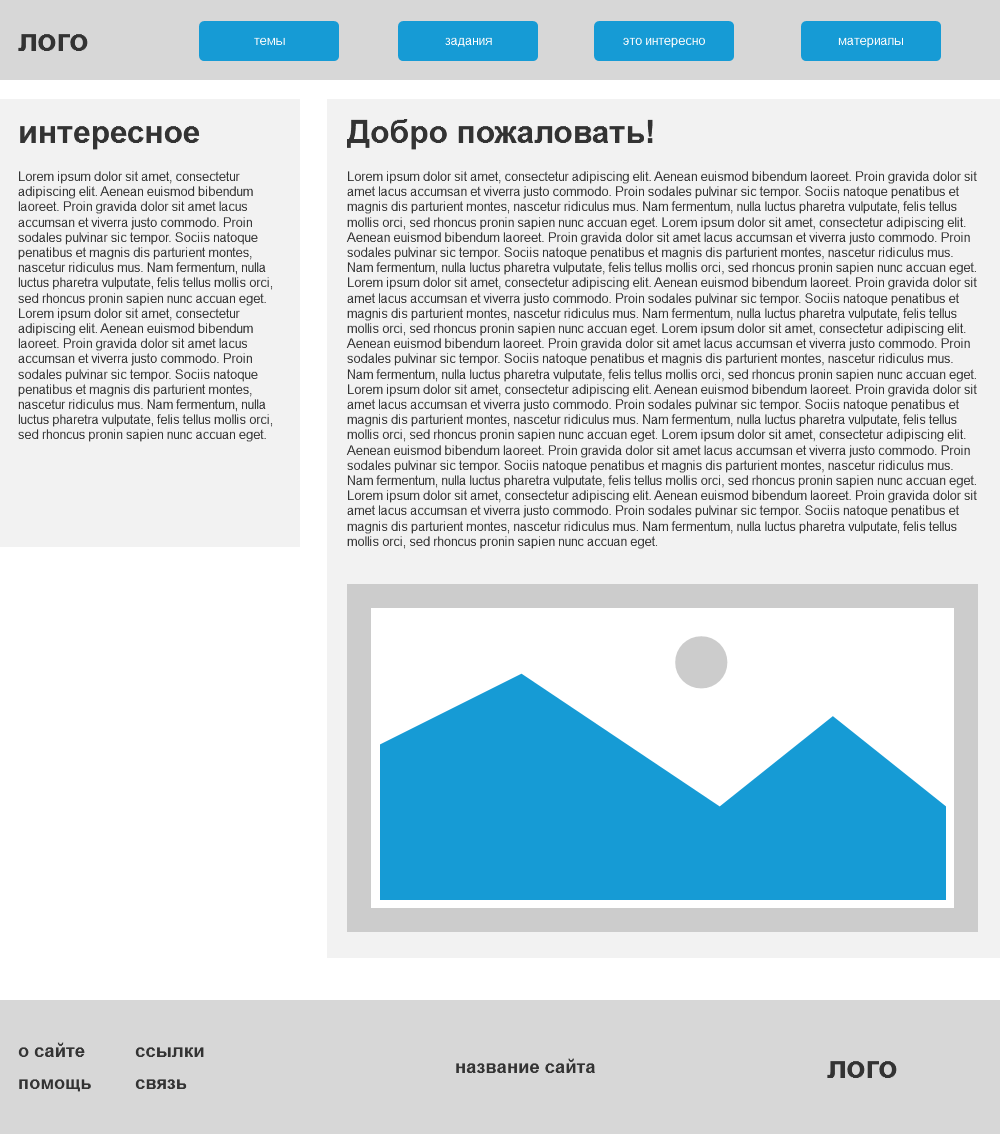


Рисунок 3.1 – Макет главной страницы

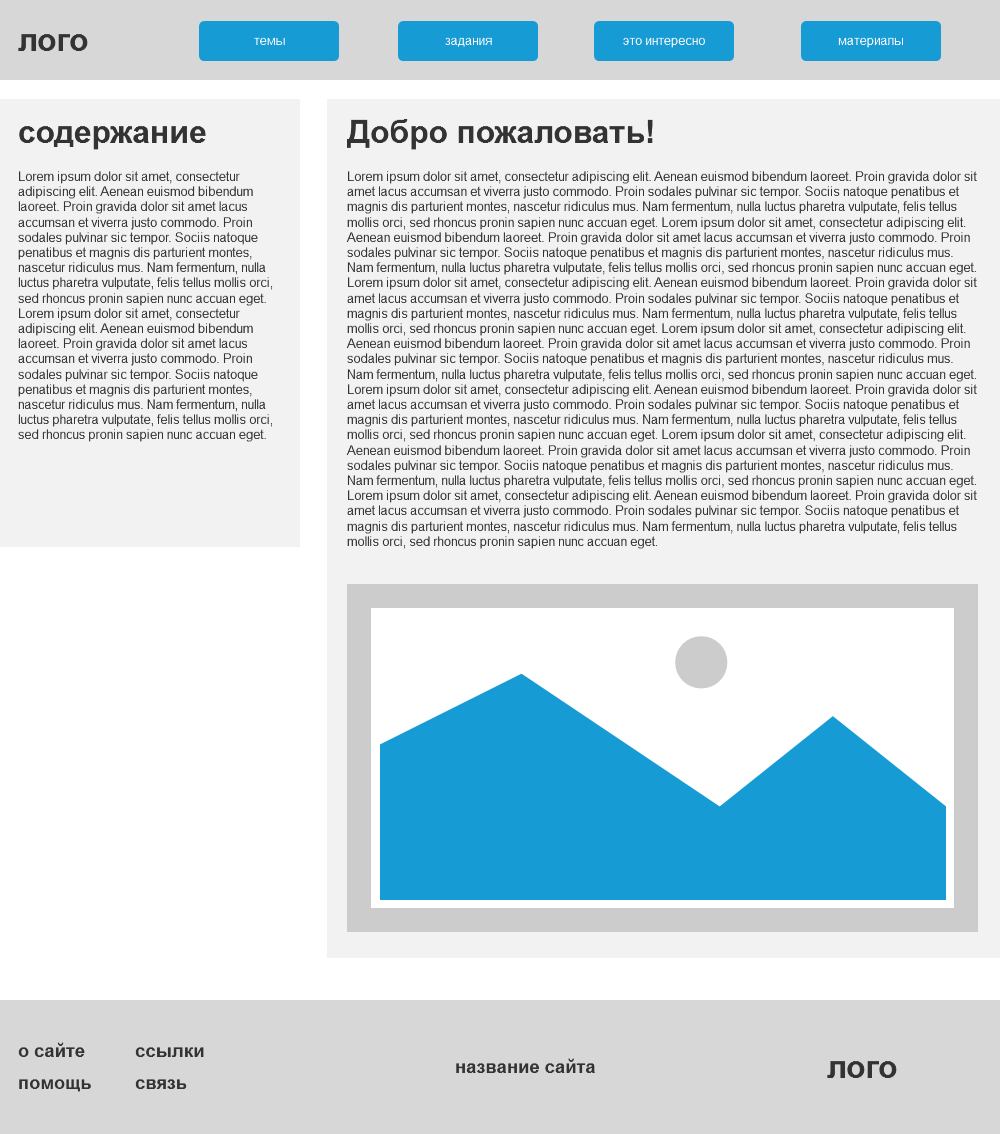


Рисунок 3.2 – Макет страницы «Темы»



Рисунок 3.3 – Макет страницы материалы

Дизайны страниц:

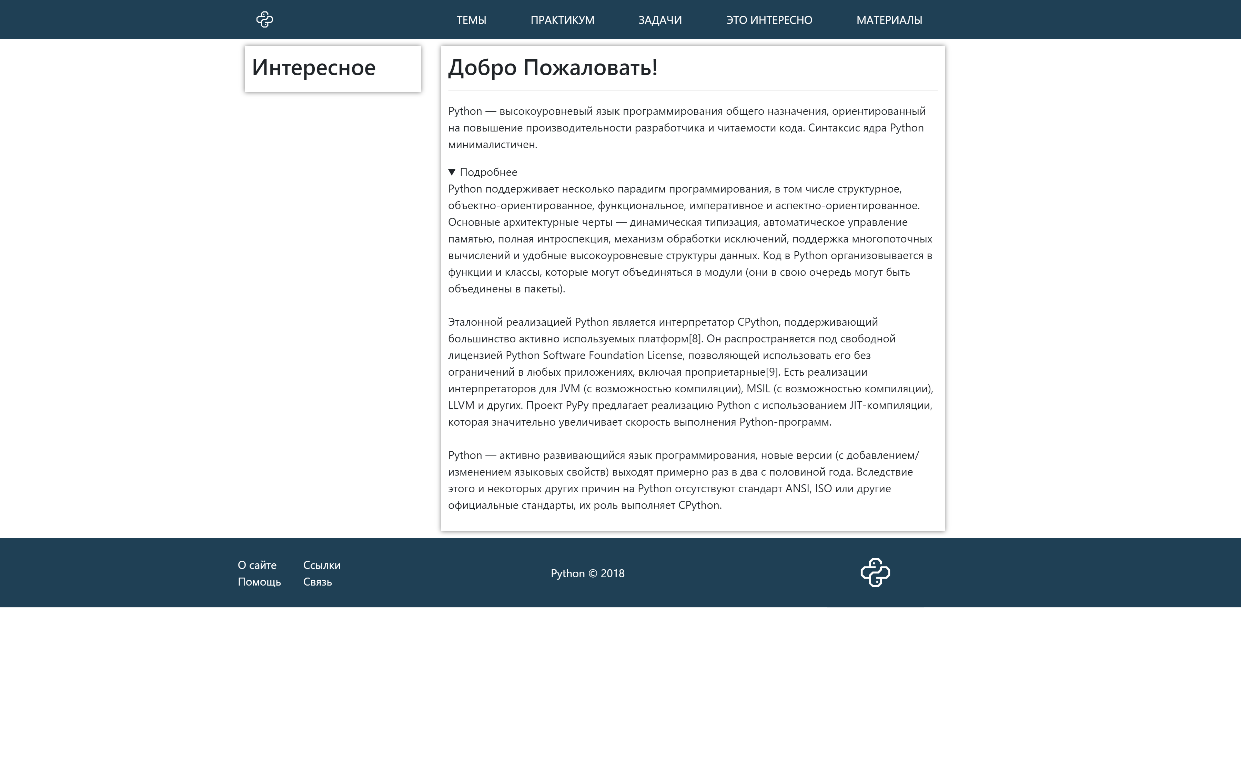


Рисунок 3.4 – Дизайн главной страницы

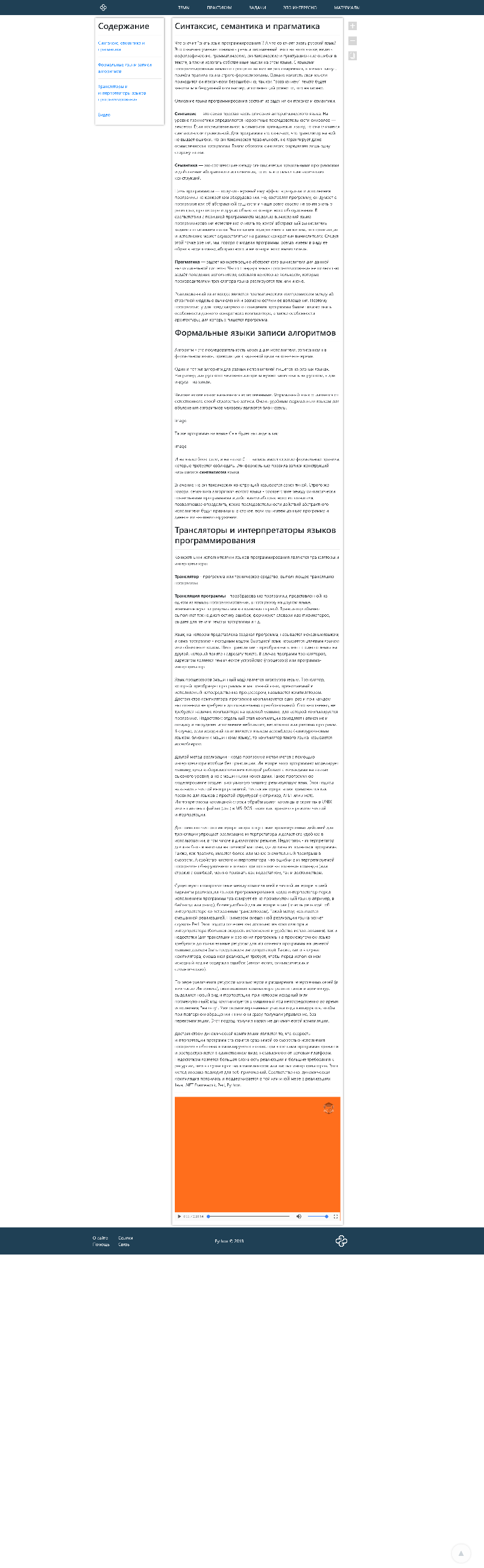


Рисунок 3.5 – Дизайн страницы «Темы»

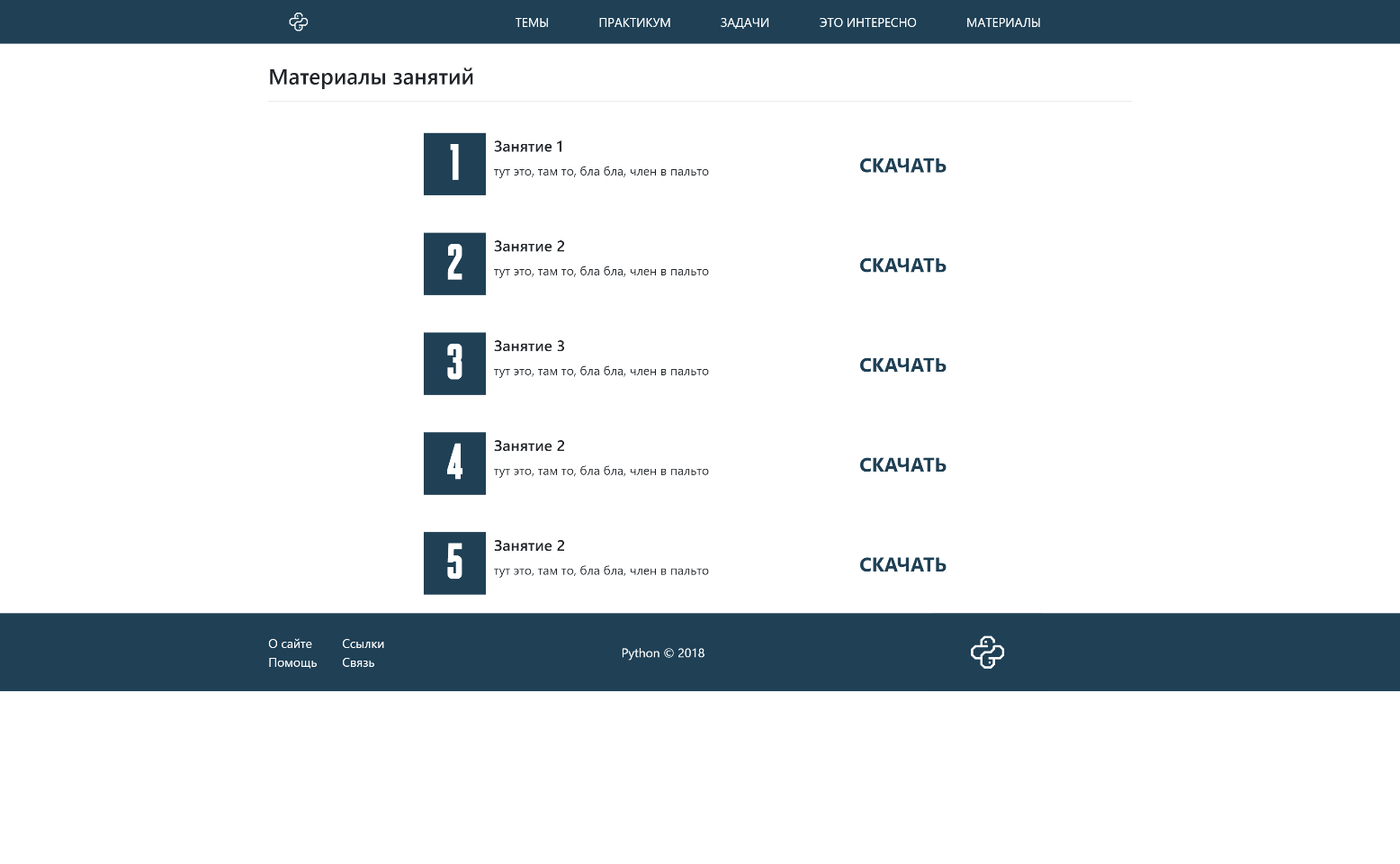


Рисунок 3.5 – Дизайн страницы «Материалы»

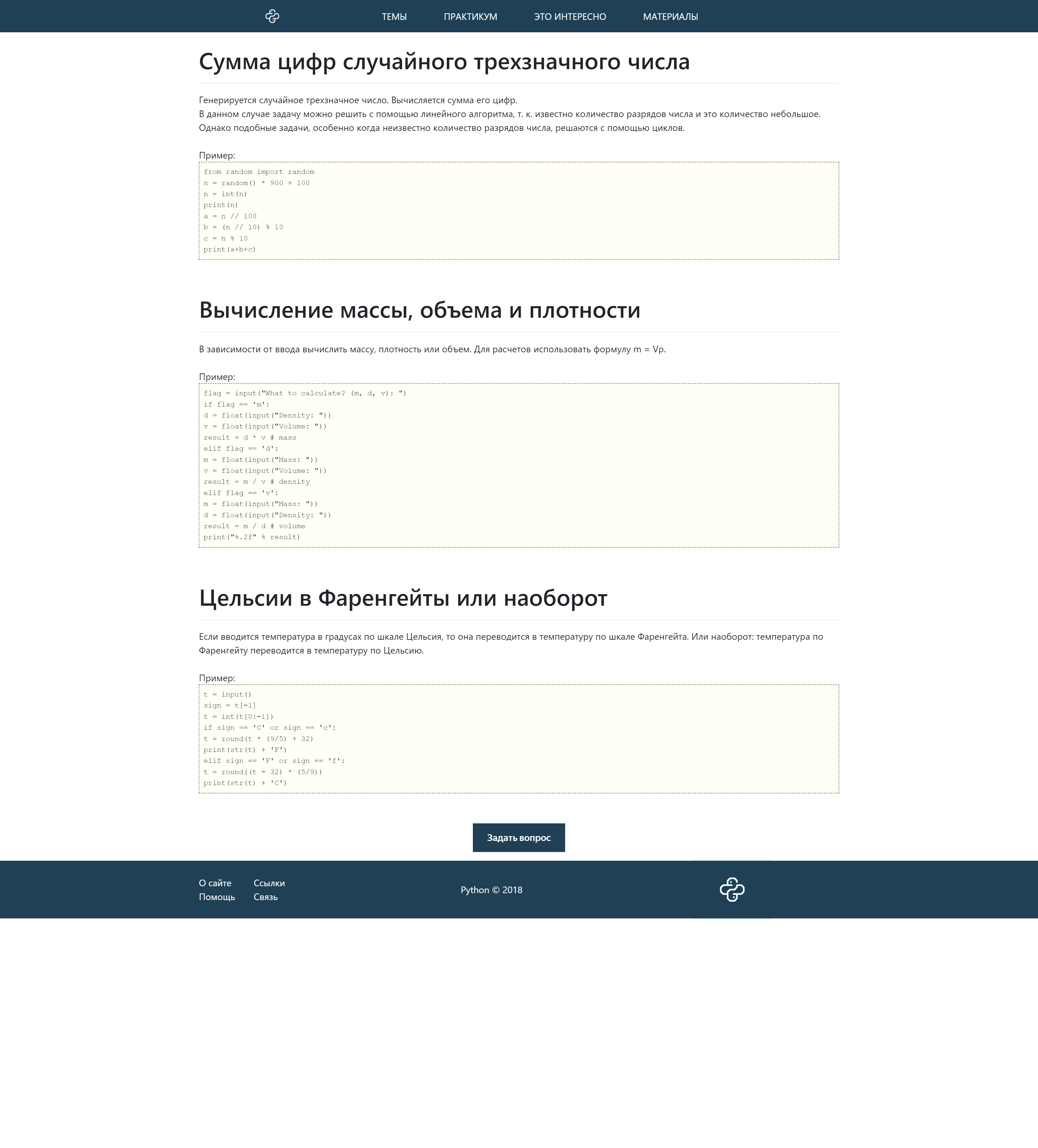


Рисунок 3.6 – Дизайн страницы «Практикум»

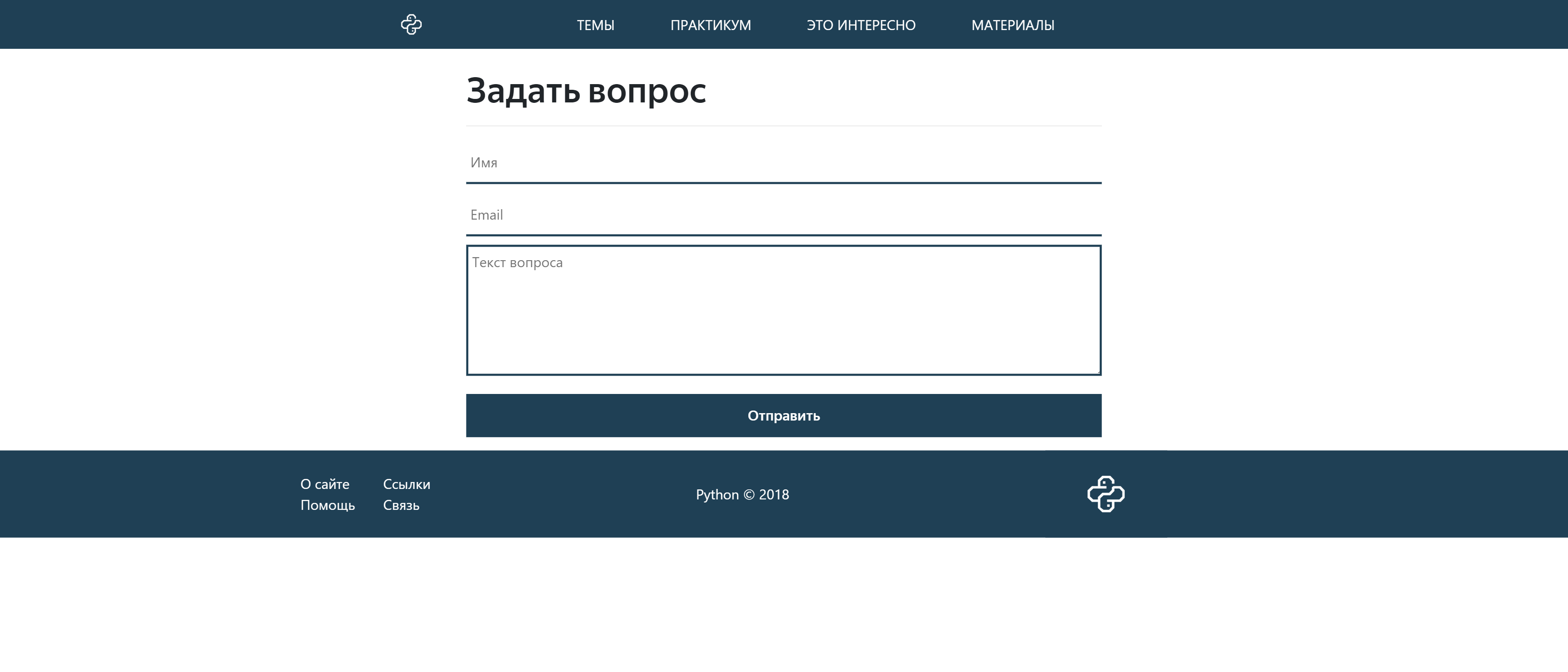


Рисунок 3.7 – Дизайн страницы «Задать вопрос»

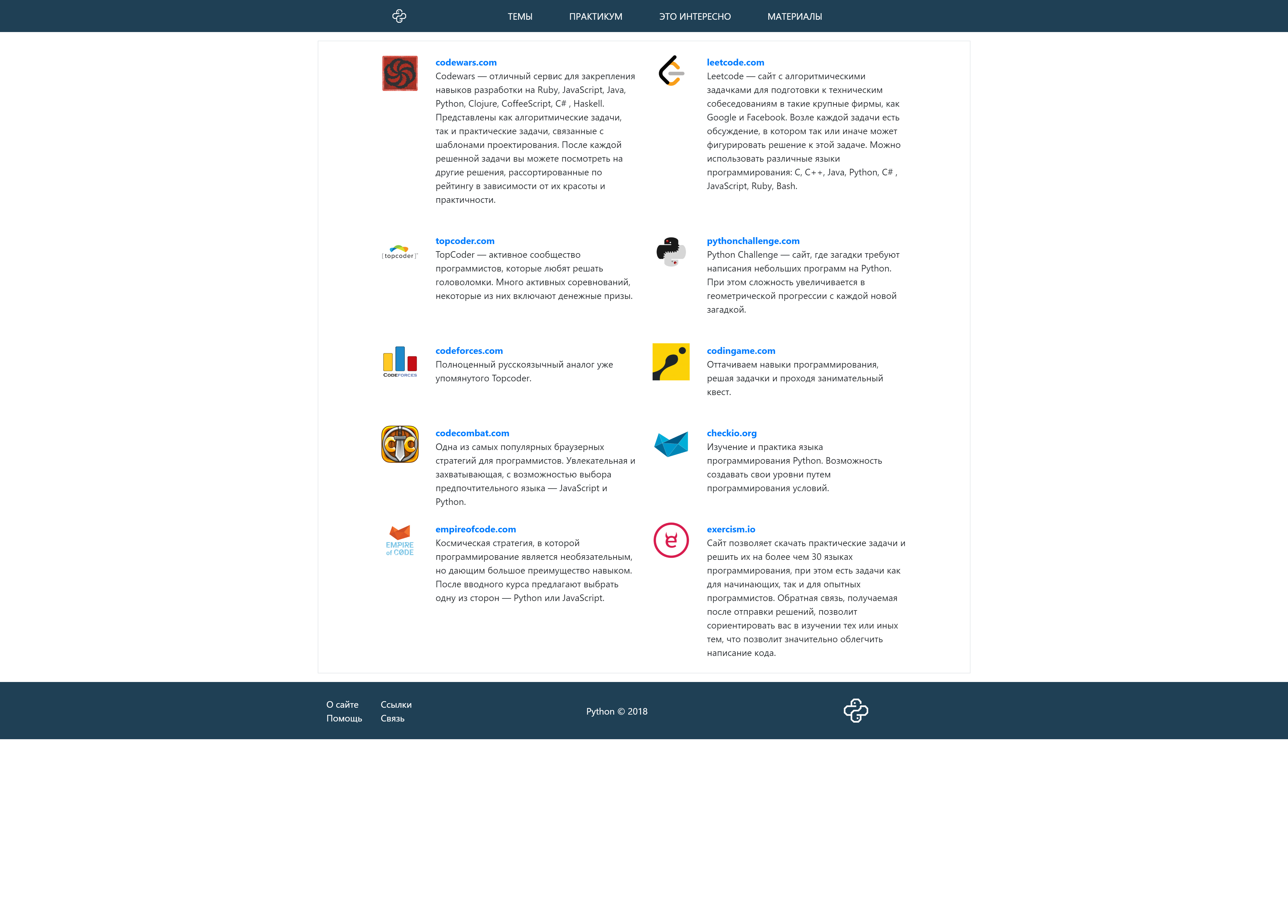


Рисунок 3.8 – Дизайн страницы «Это интересно»

# 4. Реализация

Сайт включает в себя главную страницу (Index.html), страницы с занятиями (lesson1.html), страницу с материалами (material.html), страницу с заданиями(exercise.html), страницу с интересными ссылками (interesting.html), стили для страниц (style.css), стиль для печати (print.css).

Внутри сайта есть видеоматериалы к урокам, ссылки на интересные сайты и возможность скачивать материалы к урокам.

Есть функция обратной связи, если ученику что-либо не понятно, то он может задать вопрос перейдя по ссылке под каждым уроком.

Внешний вид общей структуры:



Рисунок 4.1 – Начало документа



Рисунок 4.2 – Шапка + Меню



Рисунок 4.3 – Подвал

Рассмотрим часть контента на главной странице. Так как мы используем бутстрап, то придерживаемся некоторых правил:

Каждая секция заключается в несколько классов:

1. Container.
2. Row.
3. Col-lg.

То есть, создаем контейнер заданной шириной 1170px, с отступами по краям в 15px, далее создаем строку и в не используем класс col-lg, этот класс отвечает за количество столбцов для элементов, к которым мы это применяем, а также указываем размер, для адаптации под различные разрешения.

Сетка bootstrap имеет 12 колонок, при расположении элементов будет отталкиваться от этого числа. Рассмотрим на примере контент части:

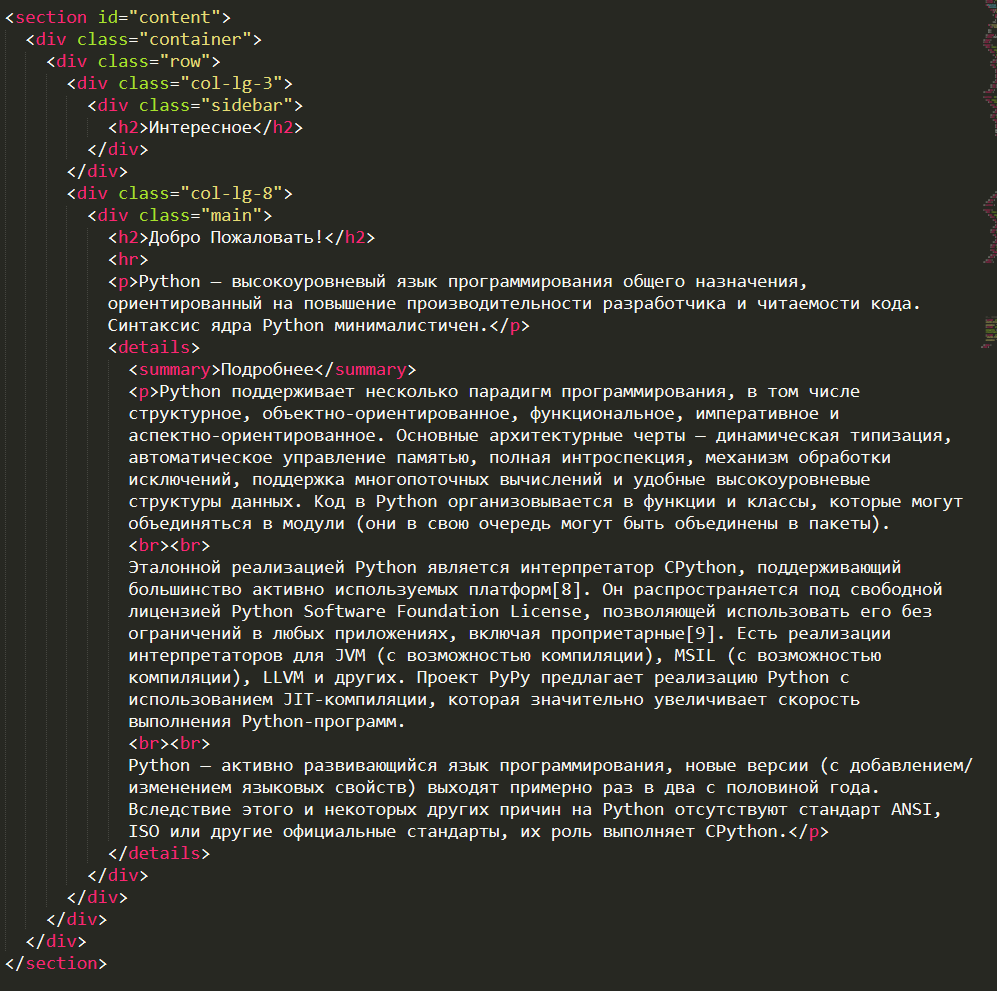


Рисунок 4.4 – Код контент части

Для блока «Интересное» мы выделяем 3 колонки из сетки, и для блока с основным текстом выделяем 8 колонок.

Также рассмотим теги details и summary. В тег details мы вставляем то, что будет под неким «спойлером», а summary отвечает за название «спойлера»

Страница «Темы» копирует главную, то есть мы имеем такую же шапку, подвал и контент, за исключением того, что вместо блока «Интересное» сделан блок «Содержание».

Страница «Материалы» также копирует шапку и подвал, но контент видоизменен:



Рисунок 4.5 – Страница материалы

Каждый блок с материалами находится в отдельном контейнере. Выделяем 1 столбец под изображение с номером, 5 столбцов под описание материалов и 2 столбца под кнопку «Скачать».

**5. Тестирование**

После завершения основных работ по созданию программной части веб-ресурса, специалист контроля качества разработки (другими словами — Тестировщик) получает всю необходимую документацию и материалы и приступает к тестированию сайта. Для организации тестирования веб-сайта предусмотрена специально разработанная методика, согласно которой и осуществляется проверка.

Тестирование по пунктам:

1) Начинается все с подготовительных работ — тестировщик изучает полученную документацию (анализирует функционал по тех. заданию, изучает конечные макеты сайта и составляет план теста для дальнейшего тестирования)

2) Функциональное тестирование — наиболее продолжительный этап проверки ресурса. Суть этого процесса заключается в проверке всего описанного функционала:

* Проверки работы всех обязательных функций сайта;
* Тестирования работоспособности пользовательских форм на сайте (например, обратная связь, добавление комментария в блог);
* Проверки работы поиска (включая релевантность результатов);
* Проверки гиперссылок, поиск нерабочих ссылок;
* Проверки подгрузки файлов на сервер;
* Проверки работоспособности счётчиков, установленных на страницах сайта;
* Просмотр на соответствие содержимого страниц сайта исходному контенту, предоставленному заказчиком.

3) Тестирования Верстки — при проверке верстки первым делом тестировщик проверяет расположения элементов, соответствие их позиций предоставленным макетам, а также проверяет оптимизацию изображений и графики. Далее осуществляется проверка валидности кода. В процессе вёрстки важно соблюдать корректную иерархию объектов, и важно удостовериться в её валидности по факту завершения работ. Браузеры, несмотря на явно неверный код, в любом случае постараются отобразить веб-страницу. Но поскольку не существует единого регламента о том, как же должен быть показан «кривой» документ, каждый браузер пытается сделать это по-своему. А это в свою очередь приводит к тому, что один и тот же документ может выглядеть по-разному в различных браузерах. Исправление явных промахов и систематизация кода приводит, как правило, к стабильному результату. Завершив проверку на валидность, специалист приступает к проверке на кроссбраузерность, т.е. проверяет работоспособность сайта в различных браузерах, а также при различных параметрах настройки экрана.

4) Usability тестирование — проводится для оценки удобства продукта в использовании, основанный на привлечении пользователей в качестве тестировщиков и анализ полученных результатов.

Несмотря на тот факт, что проработка удобства использования ресурса осуществляется в процессе составления технического задания, разработки макетов, бывают ситуации, когда полученный результат не является оптимальным. Хотя такое и происходит достаточно редко, оптимальное решение в данном случае — внести изменения в реализованный продукт.

Тестирование проводится с участием нескольких человек из целевой аудитории, так называемых респондентов. Для проведения тестирования достаточно 4-6 человек. Существует правило 80/20, которое гласит, что 20% пользователей дают 80% результата. Поэтому такое количество респондентов максимально эффективно с точки зрения экономии времени и затрат.

5) Тестирование безопасности — На данной стадии тестирования специалист проверяет — нет ли у пользователей доступа к служебным/закрытым страницам, а также проводит проверку защиты всех критически важных страниц (например, раздела администрирования сайта) от внешнего воздействия.

6) Тестирование производительности сайта — проводится с целью определения быстродействия сайта или его части под определённой нагрузкой. Тестирование производительности включает в себя такие виды тестестирования:

* Нагрузочное тестирование — простейшая форма тестирования производительности. Нагрузочное тестирование обычно проводится для того, чтобы оценить поведение сайта (или приложения) под заданной ожидаемой нагрузкой. Этой нагрузкой может быть, например, ожидаемое количество одновременно работающих пользователей на сайте, совершающих заданное число транзакций за интервал времени. Такой тип тестирования обычно позволяет получить время отклика всех самых важных бизнес-функций.
* Тестирование быстродействия — проверка скорости загрузки сайта для определения скорости отработки скриптов, загрузки изображений и контента. Этот тест проводится с целью оптимизации процесса загрузки сайта, а также определения оптимальности настроек сервера.

Обработка ошибок

На протяжении всего этапа тестирования, специалист создаёт и дополняет отчёт о выявленных им ошибках. Данный отчёт передаётся участникам проекта, после чего руководитель проекта определяет ответственного за исправление каждой из ошибок (в общем-то, часть обязанностей со временем распределяется самим тестировщиком). Далее определяется график исправления ошибок, после чего проводится повторное тестирования с целью контроля качества исправления ошибок, а также отсутствия новых. Данная процедура повторяется пока сайт не будет соответствовать спецификациям тех. задания. Именно поэтому тестирование — столь долгий процесс.

По завершению тестирования проект готов к размещению на сервере и полноценной работе, эффективно и стабильно выполняя возложенные на него бизнес-функции. Именно тестирование является гарантом спокойного сна как для заказчика, так и для команды разработчиков веб-сайта. [6]

**5.1 Валидация**

Проводим валидацию HTML и CSS на сайте при помощи сайта w3.org. Проверяем главную страницу, для этого заходим на сайт, вставляем код страницы и запускаем проверку, как мы можем видеть на скриншоте – страница валидна.



Рисунок 5.1 – Валидация главной страницы

По этому же принципу проверяем страницу «Темы», «Материалами» и css-файл сайта:



Рисунок 5.2 – Валидация страницы «Темы»

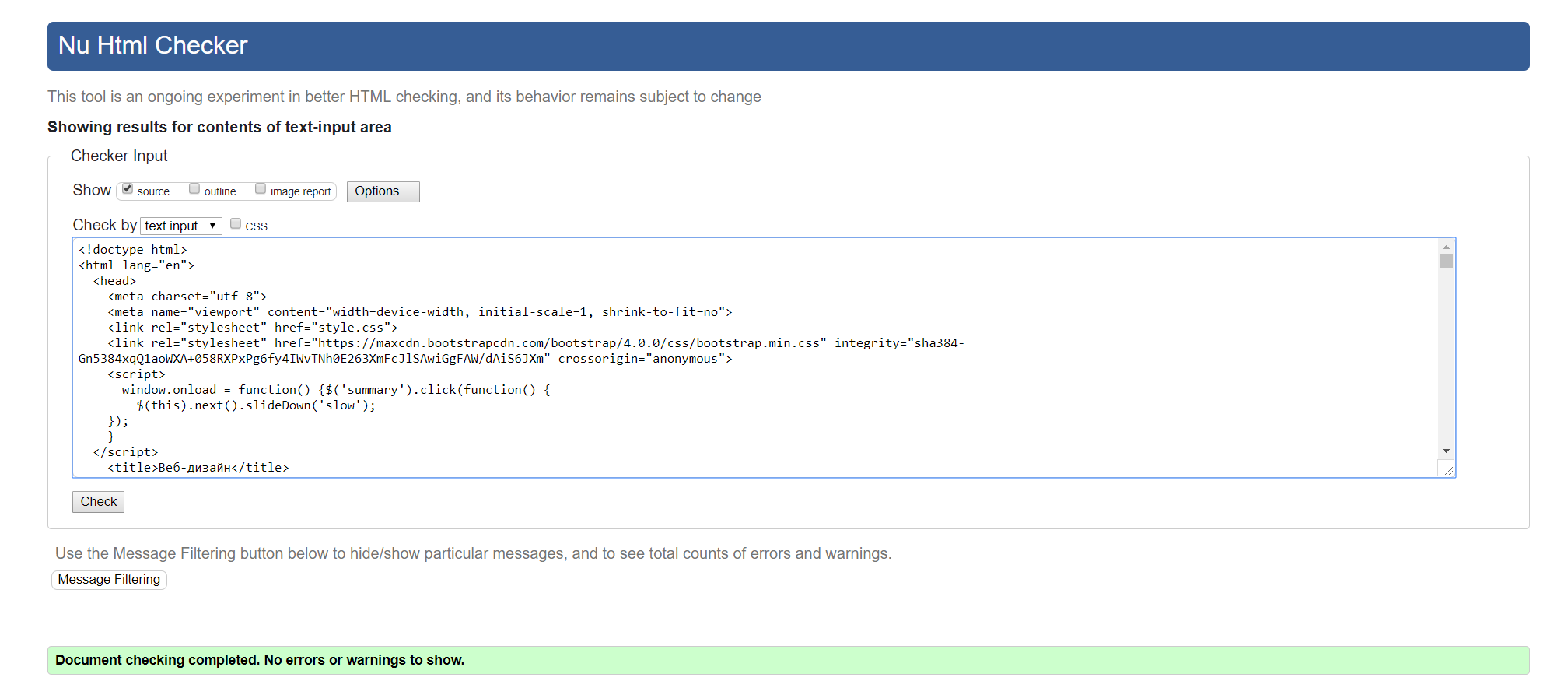


Рисунок 5.3 – Валидация страницы «Материалы»



Рисунок 5.4 – Валидация css-файла

По результатам проверки все страницы прошли проверку: ошибок не обнаружено, присутствуют лишь некоторые предупреждения.

# Заключение

В процессе создания сайта были рассмотрены различные технологии. Использован фреймворк Bootstrap.

Проработан функционал исходя из потребностей персонажей.

Проведен анализ аналогов, благодаря которому сложилось верное мнение о функциях и контенте на сайте.

В результате был получен сайт для удобного обучения языку программирования Python. С возможностью задавать вопросы квалифицированным специалистам и решать поставленные задачи.

# Список используемых источников

1. NET [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9\_%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA. Дата доступа 22.04.2018.

2. NET [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/HTML. Дата доступа 21.05.2018.

3. NET [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/CSS. Дата доступа 21.05.2018.

4. NET [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/JavaScript. Дата доступа 21.05.2018.

5. NET [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Bootstrap\_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA). Дата доступа 21.05.2018.

6. NET [Электронный ресурс] – Режим доступа: https://freshdesign.ua/blog/technology/website-testing. Дата доступа 21.05.2018.

# Приложение

**Листинг index.html**

<!doctype html>

<html lang="ru">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Поделючение стилей, библиотек бутстрапа и jquery -->

<link rel="stylesheet" href="style.css">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>

<!-- Скрипт для кнопки наверх -->

<script>

$(function() {

$(window).scroll(function() {

if($(this).scrollTop() != 0) {

$('#toTop').fadeIn();

} else {

$('#toTop').fadeOut();

}

});

$('#toTop').click(function() {

$('body,html').animate({scrollTop:0},800);

});

});

</script>

<!-- -->

<title>Python</title>

</head>

<body>

<!-- ШАПКА -->

<header>

<div class="container">

<div class="row">

<div class="col-lg-3 text-center">

<a id="maintxt" href="bootstrap.html">

<img id="logo" src="img/python.png" alt="llgoo">

</a>

</div>

<!-- МЕНЮ -->

<div class="col-lg-9">

<ul class="menu cf">

<li><a href="#">Темы</a>

<ul class="submenu">

<li><a href="lesson1.html">Занятие 1</a></li>

<li><a href="#">Занятие 2</a></li>

<li><a href="#">Занятие 3</a></li>

<li><a href="#">Занятие 4</a></li>

<li><a href="#">Занятие 5</a></li>

<li><a href="#">Занятие 6</a></li>

</ul>

</li>

<li><a href="#">Практикум</a>

<ul class="submenu">

<li><a href="exercise.html">Тема 1</a></li>

<li><a href="#">Тема 2</a></li>

<li><a href="#">Тема 3</a></li>

<li><a href="#">Тема 4</a></li>

</ul>

</li>

<li>

<a href="interesting.html">Это интересно</a>

</li>

<li>

<a href="material.html">Материалы</a>

</li>

</ul>

</div>

</div>

</div>

</header>

<section id="content">

<div class="container">

<div class="row">

<div class="col-lg-3">

<div class="sidebar">

<h2>Интересное</h2>

</div>

</div>

<div class="col-lg-8">

<div class="main">

<h2>Добро Пожаловать!</h2>

<hr>

<p>Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен.</p>

<details>

<summary>Подробнее</summary>

<p>Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

<br><br>

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ[8]. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные[9]. Есть реализации интерпретаторов для JVM (с возможностью компиляции), MSIL (с возможностью компиляции), LLVM и других. Проект PyPy предлагает реализацию Python с использованием JIT-компиляции, которая значительно увеличивает скорость выполнения Python-программ.

<br><br>

Python — активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python отсутствуют стандарт ANSI, ISO или другие официальные стандарты, их роль выполняет CPython.</p>

</details>

</div>

</div>

</div>

</div>

</section>

<footer>

<div class="container">

<div class="row">

<div class="col-lg-1 my-auto">

<ul class="my-auto">

<li>О сайте</li>

<li>Помощь</li>

</ul>

</div>

<div class="col-lg-1 my-auto">

<ul class="my-auto">

<li>Ссылки</li>

<li>Связь</li>

</ul>

</div>

<div class="col-lg-7 text-center my-auto">Python © 2018</div>

<div class="col-lg-3">

<img src="img/python.png" width="140px" height="100px" alt="">

</div>

</div>

</div>

</footer>

<!-- КНОПКА НАВЕРХ -->

<div id= "toTop" >▲</div>

</body>

</html>

**Фрагмент кода lesson1.html**

<section id="content">

<div class="container">

<div class="row">

<div class="col-lg-3">

<div class="sidebar">

<h2>Содержание</h2>

<hr>

<ul>

<li><a href="#first">Синтаксис, семантика и прагматика</a></li>

<hr>

<li><a href="#second">Формальные языки записи алгоритмов</a></li>

<hr>

<li><a href="#third">Трансляторы и интерпретаторы языков программирования</a></li>

<hr>

<li><a href="#video">Видео</a></li>

</ul>

</div>

</div>

<div class="col-lg-7">

<div class="main">

<h2 id="first">Синтаксис, семантика и прагматика</h2>

<hr>

<p>

Что значит "знать язык программирования"? А что означает знать русский язык? Это означает умение понимать речь и письменный текст на этом языке, видеть орфографические, грамматические, синтаксические и пунктуационные ошибки в тексте, а также излагать собственные мысли на этом языке. С языками программирования немного проще — на них не разговаривают, а только пишут, причём правила языка строго формализованы. Однако излагать свои мысли приходится синтаксически безошибочно, так как "осознанием" текста будет заниматься бездушный компьютер, исполняющий ровно то, что написано.

<br><br>

Описание языка программирования состоит из задания синтаксиса и семантики.

<br><br>

<b>Синтаксис</b> — это самая простая часть описания алгоритмического языка. На уровне грамматики определяются корректные последовательности символов — лексемы. Если последовательность символов принадлежит языку, то она считается синтаксически правильной. Для программы это означает, что транслятор на ней не выдает ошибки. Но синтаксическая правильность не гарантирует даже осмысленности программы. Таким образом, синтаксис определяет лишь одну сторону языка. <br><br>

<b>Семантика</b> — это соответствие между синтаксически правильными программами и действиями абстрактного исполнителя, то есть это смысл синтаксических конструкций. <br><br>

Цель программиста — получить нужный ему эффект в результате исполнения программы на конкретном оборудовании. Но, составляя программу, он думает о программе как об абстрактной сущности и чаще всего совсем не хочет знать о регистрах, процессоре и других объектах конкретного оборудования. В соответствии с позицией программиста моделью вычислений языка программирования естественно считать то, какой абстрактный вычислитель задается описанием языка. Эта позиция подкрепляется также тем, что трансляция и исполнение может осуществляться на разных конкретных вычислителях. Следуя этой точке зрения, мы, говоря о модели программы, всегда имеем в виду ее образ в виде команд абстрактного, а не конкретного вычислителя. <br><br>

<b>Прагматика</b> — задает конкретизацию абстрактного вычислителя для данной вычислительной системы. Часто стандарт языка программирования не полностью задаёт поведение исполнителя, оставляя некоторые вольности, которые производителями транслятора языка реализуются так или иначе. <br><br>

Реализованный язык всегда является прагматическим компромиссом между аб страктной моделью вычислений и возможностями ее воплощения. Поэтому программисту для предсказуемого поведения программы бывает важно знать особенности данного конкретного компилятора, а также особенности архитектуры, для которых пишется программа.

</p>

<h2 id="second">Формальные языки записи алгоритмов</h2>

<hr>

<p>

Алгоритм - это последовательность команд для исполнителя, записанная на формальном языке, приводящая к заданной цели за конечное время. <br><br>

Один и тот же алгоритм для разных исполнителей пишется на разных языках. Например, для русского человека алгоритм нужно записывать на русском, а для индуса - на хинди. <br><br>

Человеческие языки называются естественными. Формальный язык отличается от естественного своей строгостью записи. Очень удобным формальным языком для объяснения алгоритмов человеку являются блок-схемы. <br><br>

image <br><br>

Та же программа на языке С++ будет выглядеть так: <br><br>

image <br><br>

И на языке блок-схем, и на языке С++ запись имеет строгие формальные правила, которые требуется соблюдать. Эти формальные правила записи конструкций называются <b>синтаксисом</b> языка. <br><br>

Значение же синтаксических конструкций называется семантикой. Строго же говоря, семантика алгоритмического языка - соответствие между синтаксически правильными

программами и действиями абстрактного исполнителя, позволяющее определить, какие последовательности действий абстрактного исполнителя будут правильны в случае, если мы имеем данную программу и данное ее внешнее окружение. </p>

<h2 id="third">Трансляторы и интерпретаторы языков программирования</h2>

<hr>

<p>

Конкретными исполнителями языков программирования являются трансляторы и интерпретаторы. <br><br>

<b>Транслятор</b> - программа или техническое средство, выполняющее трансляцию программы. <br><br>

<b>Трансляция программы</b> - преобразование программы, представленной на одном из языков программирования, в программу на другом языке, эквивалентную по результатам выполнения первой. Транслятор обычно выполняет также диагностику ошибок, формирует словари идентификаторов, выдаёт для печати тексты программы и т.д. <br><br>

Язык, на котором представлена входная программа, называется исходнымязыком, а сама программа - исходным кодом. Выходной язык называется целевым языком или объектным кодом. Цель трансляции - преобразовать текст с одного языка на другой, который понятен адресату текста. В случае программ-трансляторов, адресатом является техническое устройство (процессор) или программа-интерпретатор. <br><br>

Язык процессоров (машинный код) является низкоуровневым. Транслятор, который преобразует программы в машинный язык, принимаемый и исполняемый непосредственно процессором, называется компилятором. Достоинство компилятора: программа компилируется один раз и при каждом выполнении не требуется дополнительных преобразований. Соответственно, не требуется наличие компилятора на целевой машине, для которой компилируется программа. Недостаток: отдельный этап компиляции замедляет написание и отладку и затрудняет исполнение небольших, несложных или разовых программ.

В случае, если исходный язык является языком ассемблера (низкоуровневым языком, близким к машинному языку), то компилятор такого языка называется ассемблером. <br><br>

Другой метод реализации - когда программа исполняется с помощью интерпретатора вообще без трансляции. Интерпретатор программно моделирует машину, цикл выборкиисполнения которой работает с командами на языках высокого уровня, а не с машинными командами. Такое программное моделирование создаёт виртуальную машину, реализующую язык. Этот подход называется чистой интерпретацией. Чистая интерпретация применяется как правило для языков с простой структурой (например, АПЛ или Лисп). Интерпретаторы командной строки обрабатывают команды в скриптах в UNIX или в пакетных файлах (.bat) в MS-DOS также как правило в режиме чистой интерпретации. <br><br>

Достоинство чистого интерпретатора: отсутствие промежуточных действий для трансляции упрощает реализацию интерпретатора и делает его удобнее в использовании, в том числе в диалоговом режиме. Недостаток - интерпретатор должен быть в наличии на целевой машине, где должна исполняться программа. Также, как правило, имеется более или менее значительный проигрыш в скорости. А свойство чистого интерпретатора, что ошибки в интерпретируемой программе обнаруживаются только при попытке выполнения команды (или строки) с ошибкой, можно признать как недостатком, так и достоинством. <br><br>

Существуют компромиссные между компиляцией и чистой интерпретацией варианты реализации языков программирования, когда интерпретатор перед исполнением программы транслирует её на промежуточный язык (например, в байт-код или p-код), более удобный для интерпретации (то есть речь идёт об интерпретаторе со встроенным транслятором). Такой метод называется смешанной реализацией. Примером смешанной реализации языка может служить Perl. Этот подход сочетает как достоинства компилятора и интерпретатора (бо́льшая скорость исполнения и удобство использования), так и недостатки (для трансляции и хранения программы на промежуточном языке требуются дополнительные ресурсы; для исполнения программы на целевой машине должен быть представлен интерпретатор). Также, как и в случае компилятора, смешанная реализация требует, чтобы перед исполнением исходный код не содержал ошибок (лексических, синтаксических и семантических). <br><br>

По мере увеличения ресурсов компьютеров и расширения гетерогенных сетей (в том числе Интернета), связывающих компьютеры разных типов и архитектур, выделился новый вид интерпретации, при котором исходный (или промежуточный) код компилируется в машинный код непосредственно во время исполнения, "на лету". Уже скомпилированные участки кода кэшируются, чтобы при повторном обращении к ним они сразу получали управление, без перекомпиляции. Этот подход получил название динамической компиляции. <br><br>

Достоинством динамической компиляции является то, что скорость интерпретации программ становится сравнимой со скоростью исполнения программ в обычных компилируемых языках, при этом сама программа хранится и распространяется в единственном виде, независимом от целевых платформ. Недостатком является бо́льшая сложность реализации и бо́льшие требования к ресурсам, чем в случае простых компиляторов или чистых интерпретаторов.

Этот метод хорошо подходит для веб-приложений. Соответственно, динамическая компиляция появилась и поддерживается в той или иной мере в реализациях Java, .NET Framework, Perl, Python.

</p>

<hr>

<video id="video" width="100%" controls="controls">

<source src="resources/Занятие 1/Занятие 1. Язык Питон.mp4" type="video/mp4">

</video>

</div>

</div>

<div class="col-lg-1">

<br>

<div class="pluss"><span id="plus-txt">+</span></div>

<br>

<div class="minus"><span id="minus-txt">–</span></div>

<br>

<div class="print" onclick="window.print();"><img id="print" src="img/stack.png" alt=""></div>

</div>

</div>

</div>

</section>

**Фрагмент кода exercise.html**

<section id="content">

<div class="container">

<br>

<div class="row">

<div class="col-lg-12">

<h1>Сумма цифр случайного трехзначного числа</h1>

<hr>

</div>

</div>

</div>

<div class="container">

<div class="row">

<div class="col-lg-12">

Генерируется случайное трехзначное число. Вычисляется сумма его цифр. <br>

В данном случае задачу можно решить с помощью линейного алгоритма, т. к. известно количество разрядов числа и это количество небольшое. Однако подобные задачи, особенно когда неизвестно количество разрядов числа, решаются с помощью циклов. <br><br>

Пример:

<p class="ex">

from random import random <br>

n = random() \* 900 + 100 <br>

n = int(n) <br>

print(n) <br>

a = n // 100 <br>

b = (n // 10) % 10 <br>

c = n % 10 <br>

print(a+b+c) <br>

</p> <br>

</div>

</div>

</div>

<div class="container">

<br>

<div class="row">

<div class="col-lg-12">

<h1>Вычисление массы, объема и плотности</h1>

<hr>

</div>

</div>

</div>

<div class="container">

<div class="row">

<div class="col-lg-12">

В зависимости от ввода вычислить массу, плотность или объем. Для расчетов использовать формулу m = Vρ. <br> <br>

Пример:

<p class="ex">

flag = input("What to calculate? (m, d, v): ") <br>

if flag == 'm': <br>

d = float(input("Density: ")) <br>

v = float(input("Volume: ")) <br>

result = d \* v # mass <br>

elif flag == 'd': <br>

m = float(input("Mass: ")) <br>

v = float(input("Volume: ")) <br>

result = m / v # density <br>

elif flag == 'v': <br>

m = float(input("Mass: ")) <br>

d = float(input("Density: ")) <br>

result = m / d # volume <br>

print("%.2f" % result) <br>

</p> <br>

</div>

</div>

</div>

<div class="container">

<br>

<div class="row">

<div class="col-lg-12">

<h1>Цельсии в Фаренгейты или наоборот</h1>

<hr>

</div>

</div>

</div>

<div class="container">

<div class="row">

<div class="col-lg-12">

Если вводится температура в градусах по шкале Цельсия, то она переводится в температуру по шкале Фаренгейта. Или наоборот: температура по Фаренгейту переводится в температуру по Цельсию. <br> <br>

Пример:

<p class="ex">

t = input() <br>

sign = t[-1] <br>

t = int(t[0:-1]) <br>

if sign == 'C' or sign == 'c': <br>

t = round(t \* (9/5) + 32) <br>

print(str(t) + 'F') <br>

elif sign == 'F' or sign == 'f': <br>

t = round((t - 32) \* (5/9)) <br>

print(str(t) + 'C') <br>

</p> <br>

</div>

</div>

</div>

<div class="container">

<div class="row">

<div class="col-lg-5"></div>

<div class="col-lg-2">

<a href="question.html">

<button class="quest">Задать вопрос</button>

</a>

</div>

</div>

</div>

</section>

**Фрагмент кода question.html**

<section id="content">

<div class="container">

<br>

<div class="row">

<div class="col-lg-2"></div>

<div class="col-lg-8">

<h1 style="font-family: YanMedium;">Задать вопрос</h1>

<hr>

<form action="#" method="post">

<input class="q\_input" type="text" placeholder="Имя"> <br>

<input class="q\_input" type="email" placeholder="Email"> <br>

<textarea class="q\_txt" placeholder="Текст вопроса"></textarea> <br>

<input class="quest" type="submit">

</form>

</div>

</div>

</div>

</section>

**Фрагмент кода interesting.html**

<section id="content">

<div class="container border">

<br>

<div class="row">

<div class="col-lg-1"></div>

<div class="col-lg-1">

<img src="img/codewars.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="https://www.codewars.com/">codewars.com</a></b> <br>

Codewars — отличный сервис для закрепления навыков разработки на Ruby, JavaScript, Java, Python, Clojure, CoffeeScript, C# , Haskell. Представлены как алгоритмические задачи, так и практические задачи, связанные с шаблонами проектирования. После каждой решенной задачи вы можете посмотреть на другие решения, рассортированные по рейтингу в зависимости от их красоты и практичности.</div>

<div class="col-lg-1">

<img src="img/LeetCode\_logo.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="https://leetcode.com/">leetcode.com</a></b> <br>

Leetcode — сайт с алгоритмическими задачками для подготовки к техническим собеседованиям в такие крупные фирмы, как Google и Facebook. Возле каждой задачи есть обсуждение, в котором так или иначе может фигурировать решение к этой задаче. Можно использовать различные языки программирования: C, C++, Java, Python, C# , JavaScript, Ruby, Bash.</div>

</div>

<br><br>

<div class="row">

<div class="col-lg-1"></div>

<div class="col-lg-1">

<img src="img/topcoder.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="https://www.topcoder.com/">topcoder.com</a></b> <br>

TopCoder — активное сообщество программистов, которые любят решать головоломки. Много активных соревнований, некоторые из них включают денежные призы.</div>

<div class="col-lg-1">

<img src="img/challneg.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="http://www.pythonchallenge.com/">pythonchallenge.com</a></b> <br>

Python Challenge — сайт, где загадки требуют написания небольших программ на Python. При этом сложность увеличивается в геометрической прогрессии с каждой новой загадкой.</div>

</div>

<br><br>

<div class="row">

<div class="col-lg-1"></div>

<div class="col-lg-1">

<img src="img/codeforces.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="http://codeforces.com/">codeforces.com</a></b> <br>

Полноценный русскоязычный аналог уже упомянутого Topcoder.</div>

<div class="col-lg-1">

<img src="img/codingame.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="https://www.codingame.com/start">codingame.com</a></b> <br>

Оттачиваем навыки программирования, решая задачки и проходя занимательный квест.</div>

</div> <br><br>

<div class="row">

<div class="col-lg-1"></div>

<div class="col-lg-1">

<img src="img/codecombat.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="https://codecombat.com/">codecombat.com</a></b> <br>

Одна из самых популярных браузерных стратегий для программистов. Увлекательная и захватывающая, с возможностью выбора предпочтительного языка — JavaScript и Python.</div>

<div class="col-lg-1">

<img src="img/checkio.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="https://checkio.org/">checkio.org</a></b> <br>

Изучение и практика языка программирования Python. Возможность создавать свои уровни путем программирования условий.</div> <br><br>

</div> <br>

<div class="row">

<div class="col-lg-1"></div>

<div class="col-lg-1">

<img src="img/empireofcode.jpg" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="https://empireofcode.com/">empireofcode.com</a></b> <br>

Космическая стратегия, в которой программирование является необязательным, но дающим большое преимущество навыком. После вводного курса предлагают выбрать одну из сторон — Python или JavaScript.</div>

<div class="col-lg-1">

<img src="img/exercism.png" width="100%" alt="">

</div>

<div class="col-lg-4"> <b><a href="http://exercism.io">exercism.io</a></b> <br>

Сайт позволяет скачать практические задачи и решить их на более чем 30 языках программирования, при этом есть задачи как для начинающих, так и для опытных программистов. Обратная связь, получаемая после отправки решений, позволит сориентировать вас в изучении тех или иных тем, что позволит значительно облегчить написание кода.</div> <br><br>

</div> <br>

</div>

</section>

**Фрагмент кода material.html**

<section id="content">

<div class="container">

<div class="row">

<div class="col-lg-12">

<br>

<h3>Материалы занятий</h3>

<hr>

</div>

</div>

</div>

<div class="container">

<br>

<div class="row">

<div class="col-lg-2"></div>

<div class="col-lg-1">

<div class="block\_dwnld">

<a id="dwnld1" href="resources/Занятие 1.rar" download="resources/Занятие 1.rar">

<div class="download">

1

</div>

</a>

</div>

</div>

<div id="pdng" class="col-lg-5 my-auto">

<h5>Занятие 1</h5>

<p>Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.</p>

</div>

<div class="col-lg-2 my-auto">

<a id="dwnld" href="resources/Занятие 1.rar" download="resources/Занятие 1.rar">Скачать</a>

</div>

</div>

<br>

</div>

<!-- 22222222 -->

<div class="container">

<br>

<div class="row">

<div class="col-lg-2"></div>

<div class="col-lg-1">

<div class="block\_dwnld">

<a id="dwnld1" href="resources/Занятие 2.rar" download="resources/Занятие 2.rar">

<div class="download">

2

</div>

</a>

</div>

</div>

<div id="pdng" class="col-lg-5 my-auto">

<h5>Занятие 2</h5>

<p>Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.</p>

</div>

<div class="col-lg-2 my-auto">

<a id="dwnld" href="resources/Занятие 2.rar" download="resources/Занятие 2.rar">Скачать</a>

</div>

</div>

<br>

</div>

<!-- 3333333 -->

<div class="container">

<br>

<div class="row">

<div class="col-lg-2"></div>

<div class="col-lg-1 my-auto">

<div class="block\_dwnld">

<a id="dwnld1" href="resources/Занятие 3.rar" download="resources/Занятие 3.rar">

<div class="download">

3

</div>

</a>

</div>

</div>

<div id="pdng" class="col-lg-5 my-auto">

<h5>Занятие 3</h5>

<p>Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.</p>

</div>

<div class="col-lg-2 my-auto">

<a id="dwnld" href="resources/Занятие 3.rar" download="resources/Занятие 3.rar">Скачать</a>

</div>

</div>

<br>

</div>

<!-- 44444444 -->

<div class="container">

<br>

<div class="row">

<div class="col-lg-2"></div>

<div class="col-lg-1">

<div class="block\_dwnld">

<a id="dwnld1" href="resources/Занятие 4.rar" download="resources/Занятие 4.rar">

<div class="download">

4

</div>

</a>

</div>

</div>

<div id="pdng" class="col-lg-5 my-auto">

<h5>Занятие 4</h5>

<p>Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.</p>

</div>

<div class="col-lg-2 my-auto">

<a id="dwnld" href="resources/Занятие 4.rar" download="resources/Занятие 4.rar">Скачать</a>

</div>

</div>

<br>

</div>

<!-- 55555555555 -->

<div class="container">

<br>

<div class="row">

<div class="col-lg-2"></div>

<div class="col-lg-1">

<div class="block\_dwnld">

<a id="dwnld1" href="resources/Занятие 5.rar" download="resources/Занятие 5.rar">

<div class="download">

5

</div>

</a>

</div>

</div>

<div id="pdng" class="col-lg-5 my-auto">

<h5>Занятие 5</h5>

<p>Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.</p>

</div>

<div class="col-lg-2 my-auto">

<a id="dwnld" href="resources/Занятие 5.rar" download="resources/Занятие 5.rar">Скачать</a>

</div>

</div>

<br>

</div>

</section>

**Листинг style.css**

\* {

margin: 0;

padding: 0;

}

@font-face {

font-family: Fred;

src: url(font/fred.ttf);

}

@font-face {

font-family: YanMedium;

src: url('font/YandexSansText-Medium.ttf');

}

@font-face {

font-family: YanRegular;

src: url('font/YandexSansText-Regular.ttf');

}

a {

text-decoration: none;

color: black;

cursor: pointer;

}

a:hover {

color: black;

text-decoration: none;

}

.slide {

position: fixed;

}

#logo {

width: 78px;

height: 56px;

}

#toTop {

width:70px;

height:70px;

border-radius: 70px;

border:1px solid #ccc;

background:#f7f7f7;

text-align:center;

padding:10px;

position:fixed;

bottom:20px;

right:50px;

box-shadow: 0 0 10px rgba(0,0,0,0.5);

cursor:pointer;

display:none;

color:#1f4055;

font-family:verdana;

font-size:30px;

}

header {

width: 100%;

height: 56px;

background-color: #1f4055;

}

footer {

width: 100%;

height: 100px;

background-color: #1f4055;

color: white;

}

li {

list-style-type: none;

}

#body {

background:

-webkit-linear-gradient(rgba(5, 610, 255, 0.4), rgba(135, 60, 255, 0.0) 80%),

-webkit-linear-gradient(-45deg, rgba(120, 155, 255, 0.9) 25%, rgba(255, 160, 65, 0.9) 76%);

}

.cf:before, .cf:after {

content:" ";

display: table;

}

.cf:after {

clear: both;

}

.cf {

\*zoom: 1;

}

.menu {

z-index: 9999;

list-style:none;

width: 100%;

width: -moz-fit-content;

width: -webkit-fit-content;

width: fit-content;

text-decoration: none;

}

.menu > li {

z-index: 9999;

background: #1f4055;

float: left;

position: relative;

-webkit-transform: skewX(25deg);

}

.menu a {

display: block;

color: #fff;

text-transform: uppercase;

text-decoration: none;

}

.menu li:hover {

background: aqua;

transition: .3s;

}

.menu > li > a {

-webkit-transform: skewX(-25deg);

padding: 1em 2em;

text-decoration: none;

color: #fff;

}

/\* Dropdown \*/

.submenu {

position: absolute;

width: 200px;

left: 50%;

margin-left: -100px;

-webkit-transform: skewX(-25deg);

-webkit-transform-origin: left top;

}

.submenu li {

background-color: #34495e;

position: relative;

overflow: hidden;

}

.submenu > li > a {

padding: 1em 2em;

text-decoration: none;

color: white;

}

.submenu > li::after {

content:'';

position: absolute;

top: -125%;

height: 100%;

width: 100%;

box-shadow: 0 0 50px rgba(0, 0, 0, .9);

}

/\* Odd stuff \*/

.submenu > li:nth-child(odd) {

-webkit-transform: skewX(-25deg) translateX(0);

}

.submenu > li:nth-child(odd) > a {

-webkit-transform: skewX(25deg);

}

.submenu > li:nth-child(odd)::after {

right: -50%;

-webkit-transform: skewX(-25deg) rotate(3deg);

}

/\* Even stuff \*/

.submenu > li:nth-child(even) {

-webkit-transform: skewX(25deg) translateX(0);

}

.submenu > li:nth-child(even) > a {

-webkit-transform: skewX(-25deg);

}

.submenu > li:nth-child(even)::after {

left: -50%;

-webkit-transform: skewX(25deg) rotate(3deg);

}

/\* Show dropdown \*/

.submenu, .submenu li {

visibility: hidden;

}

.submenu li {

transition: .2s ease -webkit-transform;

}

.menu > li:hover .submenu, .menu > li:hover .submenu li {

visibility: visible;

transition: .8s;

}

.menu > li:hover .submenu li:nth-child(even) {

-webkit-transform: skewX(25deg) translateX(15px);

}

.menu > li:hover .submenu li:nth-child(odd) {

-webkit-transform: skewX(-25deg) translateX(-15px);

}

.sidebar {

margin: 10px;

padding: 10px;

box-shadow: 0 0 10px rgba(0,0,0,0.5);

width: 100%;

height: auto;

}

.main {

margin: 10px;

padding: 10px;

box-shadow: 0 0 10px rgba(0,0,0,0.5);

width: 100%;

height: auto;

}

.pluss {

position: relative;

width: 32px;

height: 32px;

background-color: #D6D6D6;

color: white;

font-size: 37px;

}

.pluss:hover {

background-color: #1f4055;

transition: .4s

}

#plus-txt {

position: absolute;

top: -15px;

right: 3px;

cursor: pointer;

}

.print {

position: relative;

width: 32px;

height: 32px;

background-color: #D6D6D6;

}

.print:hover {

background-color: #1f4055;

transition: .4s;

}

#print {

position: absolute;

top: 3.5px;

left: 3px;

}

.minus {

position: relative;

width: 32px;

height: 32px;

background-color: #D6D6D6;

color: white;

font-size: 37px;

}

.minus:hover {

background-color: #1f4055;

transition: .4s;

}

#minus-txt {

position: absolute;

top: -16px;

right: 6px;

cursor: pointer;

}

summary {

outline: none;

}

summary>p {

font-size: 14px;

}

button#plus {

width: 40px;

height: 40px;

text-decoration: none;

outline: none;

background-color: transparent;

border-image: linear-gradient(#ff3000, #ed0200, #ff096c, #d50082);

border-image-slice: 1;

}

button#min {

width: 40px;

height: 40px;

text-decoration: none;

outline: none;

background-color: transparent;

border-image: linear-gradient(#ff3000, #ed0200, #ff096c, #d50082);

border-image-slice: 1;

margin-top: 10px;

}

.download {

width: 80px;

height: 80px;

background-color: #1f4055;

float: left;

margin-right: 10px;

font-family: Fred;

color: white;

font-size: 60px;

text-align: center;

cursor: pointer;

transition: .6s;

}

.download:hover {

background-color: #fff;

color: black;

transition: .6s;

}

a#dwnld {

text-decoration: none;

color: #1f4055;

text-transform: uppercase;

font-size: 25px;

font-weight: bold;

transition: .6s;

}

.block\_dwnld {

padding-left: 10px;

}

#dwnld1:hover {

font-size: 20px;

transition: .6s;

}

#dwnld:hover {

font-size: 20px;

transition: .6s;

padding-left: 9px;

}

#pdng {

padding-left: 20px;

}

.ex {

border: dashed 1px #634F36;

background: #fffff5;

font-family: "Courier New", Courier, monospace;

padding: 7px;

font-size: 80%;

margin: 0 0 1em;

}

.quest {

width: 100%;

height: 50px;

margin-bottom: 15px;

margin-top: 15px;

background-color: #1f4055;

border: none;

outline: none;

transition: .3s;

color: white;

font-family: YanMedium;

}

.quest:hover {

background-color: #34495e;

transition: .3s;

border: 1px solid #1f4055;

}

.q\_input {

width: 100%;

height: 50px;

margin-bottom: 10px;

border: none;

border-bottom: 2px solid #1f4055;

padding-left: 5px;

outline: none;

}

.q\_txt {

width: 100%;

height: 150px;

border: 2px solid #1f4055;

outline: none;

padding: 5px;

}

.border {

margin-top: 15px;

margin-bottom: 15px;

}

@media (max-width: 768px) {

.menu {

margin-top: -43px;

float: right;

font-size: 16px;

text-align: center;

}

.menu > li > a {

padding: 0.5em 0.5em;

text-align: center;

}

.submenu > li > a {

padding: 0.5em 0.5em;

}

.menu > li {

-webkit-transform: skewX(0deg);

}

.menu > li > a {

-webkit-transform: skewX(0deg);

}

.submenu {

-webkit-transform: skewX(0deg);

}

.submenu > li::after {

height: 100%;

width: 100%;

box-shadow: none;

}

.submenu > li:nth-child(odd) {

-webkit-transform: skewX(0deg) translateX(0);

}

.submenu > li:nth-child(odd) > a {

-webkit-transform: skewX(0deg);

}

.submenu > li:nth-child(odd)::after {

right: 0%;

-webkit-transform: skewX(0deg) rotate(0deg);

}

.submenu > li:nth-child(even) {

-webkit-transform: skewX(0deg) translateX(0);

}

.submenu > li:nth-child(even) > a {

-webkit-transform: skewX(0deg);

}

.submenu > li:nth-child(even)::after {

left: 0%;

-webkit-transform: skewX(0deg) rotate(0deg);

}

.submenu li {

transition: .2s ease -webkit-transform;

}

.menu > li:hover .submenu li:nth-child(even) {

-webkit-transform: skewX(0deg) translateX(0px);

}

.menu > li:hover .submenu li:nth-child(odd) {

-webkit-transform: skewX(0deg) translateX(0px);

}

.pluss, .minus, .print {

display: none;

}

}

@media (max-width: 480px) {

header {

height: 120px;

}

.sidebar {

margin-left: 0px;

margin-right: 0px;

}

.main {

margin-left: 0px;

margin-right: 0px;

}

.menu {

margin: 0 auto;

margin-top: 14px;

float: none;

font-size: 14px;

text-align: center;

}

.menu > li > a {

padding: 0.5em 0.5em;

}

.submenu > li > a {

padding: 0.5em 0.5em;

}

.menu > li {

-webkit-transform: skewX(0deg);

}

.menu > li > a {

-webkit-transform: skewX(0deg);

}

.submenu {

-webkit-transform: skewX(0deg);

}

.submenu > li::after {

top: 0%;

width: 200%;

box-shadow: none;

border-bottom: 1px solid white;

}

.submenu > li:nth-child(odd) {

-webkit-transform: skewX(0deg) translateX(0);

}

.submenu > li:nth-child(odd) > a {

-webkit-transform: skewX(0deg);

}

.submenu > li:nth-child(odd)::after {

-webkit-transform: skewX(0deg) rotate(0deg);

}

.submenu > li:nth-child(even) {

-webkit-transform: skewX(0deg) translateX(0);

}

.submenu > li:nth-child(even) > a {

-webkit-transform: skewX(0deg);

}

.submenu > li:nth-child(even)::after {

-webkit-transform: skewX(0deg) rotate(0deg);

}

.submenu li {

transition: .2s ease -webkit-transform;

}

.menu > li:hover .submenu li:nth-child(even) {

-webkit-transform: skewX(0deg) translateX(0px);

}

.menu > li:hover .submenu li:nth-child(odd) {

-webkit-transform: skewX(0deg) translateX(0px);

}

.pluss, .minus, .print {

display: none;

}

}

@media print {

p {

font-size: 20px;

}

video {

display: none;

}

ul {

display: none;

}

.pluss {

display: none;

}

.minus {

display: none;;

}

#logo {

display: none;

}

#foot\_logo {

display: none;

}

footer {

width: 0px;

height: 0px;

}

.sidebar {

display: none;

}

header {

width: 0px;

height: 0px;

}

.print {

display: none;

}

#maintxt {

display: none;

}

.b {

display: none;

}

}