КУРСОВАЯ РАБОТА

на тему: Разработка одностраничного веб-приложения с использованием HTML, CSS и JavaScript

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2025 г.

# Содержание

Введение.................................................................................3

Глава 1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**................................4

1.1 HTML............................................................................4

1.2 CSS..............................................................................6

1.3 JavaScript......................................................................8

Глава 2 **ПРАКТИЧЕСКАЯ ЧАСТЬ**..............................10

2.1 **Описание предметной области**....................................................................10

2.2 Структура проекта......................................................................11

2.3 Реализация................................................................................12

2.4 Тестирование..............................................................................15

2.5 Возможности доработки.................................................................17

Заключение.....................................................................................18

Список использованных источников........................................................19

Приложения....................................................................................20

# Введение

В наше время Интернет - это одно из наиболее важных средств

массовой информации и коммуникации, которое находится по своей значимости в одном ряду с прессой, телефоном и телевидением. Начиная с 1969 года, мировая сеть - Интернет стремительно развивалась, в наше время превратилась в одно из основных средств, используемых для общения, развлечений, рекламы, торговли и источник разнообразной полезной информации для любой категории пользователей. Быстро растет количество изданий, посвященных сети Интернет, что предвещает широкое ее распространение даже в далеких от техники областях.

На сегодняшний день любой человек, у которого есть компьютер, может загрузить с его помощью новости, иллюстрации, видео - или аудиоинформацию по любой интересующей его теме. Он может получать эту информацию тогда, когда он этого пожелает.

Создание Web-сайта - это для любой компании коммерческая необходимость и условие выживаемости. При такой скорости развития Интернета, ни одна компания, поставившая перед собой цель поступательного развития, не сможет обойтись без сайта. Интернет сейчас - это самое популярное средство массовой информации, а создание сайтов - серьезный инструмент решения маркетинговых стратегий, способных действенно и наиболее эффективно с точки зрения финансовой привлекательности, технических возможностей и в рекламных целях использовать достижения мировой компьютерной сети.

Предусмотреть на сайте возможность просмотра новостей, ознакомление с благотворительностью, знакомство с различными услугами, узнать о семинарах, консультациях и конечно же с фото/видео галереей. Необходимо предусмотреть подходящий дизайн, интерфейс должен быть понятен даже не много опытному пользователю.

Безусловно, одним из важнейших требований является красивый, приятный и самое главное простой дизайн сайта, который бы не раздражал глаз пользователя и гармонично смотрелся в совокупности со всеми предоставляемыми функциями.

Объектом исследования является процесс разработки web-сайта. Предмет – Разработка сайта HTML и CSS.  
Цель работы – Разработка сайта web-сайта «Ромашки».  
Задачи исследования:

1. Ознакомиться с современными Интернет-технологиями;  
2. Изучить основные понятия и программный инструментарий, применяемый для разработки и создания Web-сайтов;  
3. Ознакомиться с основными правилами и рекомендациями по разработке и созданию Web-сайтов;  
4. Разработать структуру сайта;  
5. Представить пошаговую стратегию разработки web-сайта. Методы исследования: изучение литературы, анализ, синтез, классификация и обобщение.

# Глава 1. Теоретические основы HTML, CSS и JavaScript

## 1.1 HTML

HTML (от английского HyperText Markup Language) — это язык гипертекстовой разметки страницы. Он используется для того, чтобы дать браузеру понять, как нужно отображать загруженный сайт.

Язык состоит из тегов — это своеобразные команды, которые преобразовываются в визуальные объекты в браузере пользователя. Например, тег <img> используется для размещения изображений на странице. У него есть обязательный атрибут src, в котором указывается ссылка на файл.

Язык HTML существует в нескольких вариантах и продолжает развиваться, но конструкции HTML скорее всего будут использоваться и в дальнейшем. Изучая HTML и познавая его глубже, создавая документ в начале изучения HTML и расширяя его насколько это возможно, мы имеем возможность создавать Web-страницы, которые могут быть просмотрены многими брfузерами Web, как сейчас, так и в будущем.

Простыми словами, HTML — это каркас сайта. В нем могут быть прописаны:

* ●  ссылки;
* ●  таблицы;
* ●  изображения;
* ●  блоки;
* ●  абзацы;
* ●  формы;
* ●  заголовки и так далее.

HTML язык содержит несколько основных компонентов, которые выполняют определенные функции:  
1. Тег (дескриптор). HTML полностью состоит из дескрипторов.

Они посылают команды веб-браузеру, выполняющему конкретные

5

функции. К примеру, вставляет картинки или создает списки.  
2. Атрибут (аргумент). Предназначен для внесения изменений в

теге. То есть, он может выровнять по краю или центру абзац/картинку внутри самого тега.

3. Значение. Определяет изменения от атрибутов.  
Например, при выравнивании текста или изображения, для атрибута можно задать различные значения: текстовые (left либо right), или числовые (размеры картинки по ширине и высоте).

Посетив в браузере любую web-страницу, вы можете посмотреть, как она выглядит в кодовом формате. Можно заметить, что в начале любого документа в формате HTML есть строка , обозначающая версию HTML и порядок написания и применения тегов.  
За ним следует непосредственно документ страницы, имеющий  
начало и конец, которые обозначаются открывающимся (<html>) и закрывающимся (</html>) тегами соответственно. Этот парный тег является контейнером, где находится все содержимое web-страницы – весь полезный контент охвачен этими двумя элементами.

Внутри самого документа находятся 2 блока: <head> и <body>, и выглядит это следующим образом:  
<html>  
<head></head>

<body></body>  
</html>  
В <head> содержатся служебные элементы для поисковиков и браузеров, а все видимые пользователю элементы содержаться в тег <body>.

HTML - язык разметки документов основанный на тэгах. Документ на языке HTML представляет собой набор элементов, при этом начало и конец каждого элемента обозначается служебными символами - тегами. Все тэги HTML начинаются с «<« (левой угловой скобки) и заканчиваются символом «>« (правой угловой скобки). Завершающий тег выглядит также, как начальный, и отличается от него прямым слэшем перед текстом внутри угловых скобок.

**<HTML></HTML>**

HTML регистронезависимый язык, теги могут быть написаны как строчными, так и заглавными буквами (в отличие от XHTML). Теги могут быть вложенными друг в друга.

<**HTML>**

**<HEAD>**

**<TITLE>**

Заголовок страницы

**</TITLE>**

**</HEAD>**

**</HTML>**

Тэги могут быть пустыми, то есть не содержать текста или других вложенных конструкций (например, <**br>**который переводит строку). Закрывающий тег в таком случае не указывается.

Также, элементы разметки могут иметь атрибуты, задающие их свойства (например, размер шрифта, цвет, расположение). Атрибуты задаются в начале тега.

**<ahref=«**http://www.yandex.ru**»>**Пример элемента с атрибутом **href.</a>**

Теги можно разделить на следующие группы.

*Гиперссылки*

**<AHREF=«**filename**»target=«\_self»>**текстссылки**</A>**

где filename - имя файла (может быть и локальным) или адрес страницы вInternet, на который нужно совершить переход.

текст ссылки - текст гипертекстовой ссылки, который будет отображаться в браузере, как правило, выделяется подчеркиванием.

target - задает окно или фрейм, в котором будет открыт документ, при переходе по ссылке. Он может принимать значения:

\_top - документ откроется в текущем окне

\_blank - документ откроется в новом окне

\_self - документ откроется в текущем фрейме

\_parent - документ откроется в родительском фрейме

По умолчанию принимает значение \_self.

*Текстовые ссылки.*

**<H1></H1>**,**<H2></H2>**, … ,**<H6></H6>** - заголовки 1-6 уровней. Применяются для выделения частей выводимого текста (заголовок 1 - будет выведен очень большим, 6 - будет размером сопоставимым с обычным текстом).

**<P>** - обозначает начало нового абзаца. Закрывающий тег**</P>**, не является необязательным.

**<BR>** - переход на новую строку. Закрывающий тег**</br>**, отсутствует.

**<HR>** - горизонтальная линия.

**<BLOCKQUOTE></BLOCKQUOTE>** - цитата. Выделение заданного текста как цитаты.

**<PRE></PRE>** - режим предпросмотра. При этом текст заключается в рамку и выводиться не форматированным (то есть все теги, кроме </PRE>, игнорируются, но при этом переводы строки ставятся там, где они присутствуют в исходном документе).

**<DIV></DIV>** - блок текста (как правило, применяется для использования каскадных стилей CSS).

**<SPAN></SPAN>** - строка (как правило, применяется для использования каскадных стилей CSS).

## 1.2 CSS

Аббревиатура CSS расшифровывается как Cascading Style Sheets, что

в переводе означает «каскадные таблицы стилей». Это язык разметки, используемый для визуального оформления веб-сайтов.

Объекты, расположенные на странице, размещаются с помощью HTML. А вот CSS отвечает за то, как эти объекты выглядят. Их размер, цвет, фоновое изображение, степень прозрачности, расположение относительно других элементов, поведение при наведении курсора, визуальное изменение кнопок при нажатии и т.п.

Язык CSS быстро стал стандартом в веб-разработке, потому что он позволяет быстро изменить визуальное оформление сайта, не прибегая к использованию более сложных языков программирования.

Достаточно ознакомиться с простейшими правилами CSS, и можно легко собрать вполне симпатичный сайт со всем необходимым контентом. «Простота» обеспечивается за счет понятного синтаксиса.

Синтаксис CSS в отдельном файле в соответствующем формате (.css) выглядит так:

}

Селектор – это ссылка на элемент в HTML, над которым будет вестись работа (оформление).

Свойство – определенная характеристика элемента, которую нужно изменить. Например, размер или цвет.

CSS-файл может быть подключен на страницу при помощи тега

<link> , который должен находиться в секции <head> HTML-страницы – Это самый часто используемый способ подключения CSS.  
Внутренние таблицы стилей встраиваются в HTML-страницу, в секцию <head> , и находятся они внутри тегов <style>. И последний способ

подключения CSS – это встроенные стили. Они пишутся непосредственно внутри HTML-тегов в атрибуте style. Такие стили, действуют только на тот тег, в котором написаны.

## 1.3 JavaScript

JavaScript — это интерпретируемый язык программирования, который используют для написания frontend- и backend-частей сайтов, а также мобильных приложений. Часто в текстах и обучающих материалах название языка сокращают до JS. Это язык программирования высокого уровня, то есть код на нем понятный и хорошо читается.

JS поддерживают все популярные браузеры. Во frontend-части сайтов язык используют для создания интерактива (анимаций, всплывающих форм, автозаполнения), так как он связан с [HTML](https://blog.skillfactory.ru/glossary/html/) и [CSS](https://blog.skillfactory.ru/glossary/css/) и может ими манипулировать. В backend-части с языком JavaScript работают на платформе Node.js. С ее помощью, например, разрабатывают серверные веб-приложения и подключают библиотеки. В поисковике Google на JavaScript работает строка автозаполнения, а Netflix, Uber, eBay используют его в своем backend. Уже 6 лет JS — [самый популярный](https://incrussia.ru/news/yazyki-programmirovaniya/) язык среди разработчиков по версии [GitHub](https://blog.skillfactory.ru/glossary/github-2/).

JavaScript простыми словами называют языком скриптов или сценариев. Скрипты — это набор инструкций, которые выполняются при загрузке страницы. Браузер самостоятельно интерпретирует код на JavaScript, для этого даже не требуется компиляция (перевод языка программирования в машинный код).

Скрипты можно прописать внутри кода страницы или подключить к HTML отдельным файлом. Например в стандартной разметке index.html JS-код прописывают внутри тега script, помещенного в тег body:

**<!DOCTYPE html>**

<**html** lang="en">

<**head**>

<**meta** charset="UTF-8">

<**meta** name="viewport" content="width=device-width, initial-scale=1.0">

<**title**>Документ</**title**>

</**head**>

<**body**>

<**h1**>Привет!</**h1**>

<**script**>

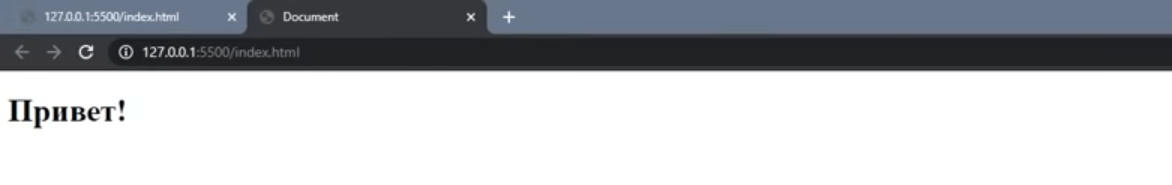
console.log('Привет, мир!');

</**script**>

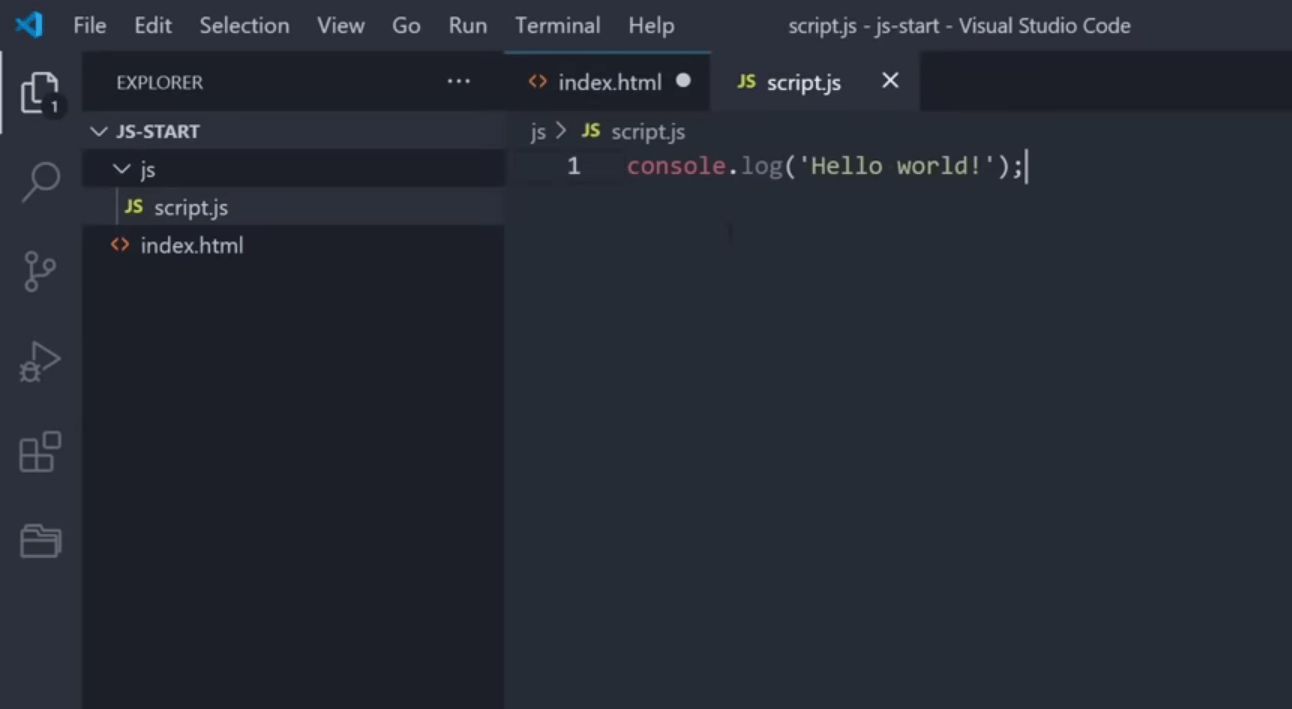
</**body**>

</**html**>

Тогда в браузере благодаря инструкции console.log появится слово «Привет!».



Но обычно для этого создается отдельный файл с расширением .js. Его называют script.js и прописывают код в нем:



JS — мультипарадигменный язык, так как он поддерживает разные парадигмы (стили) программирования и особенности:

* **Объектно-ориентированный** — представляет программу как систему отношений между объектами — наборами данных и функций.
* **Императивный** — в этом стиле программирования используют инструкции, которые выполняются последовательно, а для доступа к данным используются переменные, значение которых может быть переписано в ходе вычисления.
* **Функциональный** — в нем вычисление основано на математических функциях. Так как функциональное программирование не опирается на переменные, которые могут менять значение, результаты запросов будут всегда одинаковыми.
* **Динамическая типизация**: Переменные в JavaScript не требуют объявления типа данных. Они могут изменять свой тип в процессе выполнения, что делает код более гибким, но также требует внимательного управления типами.
* **Обработка событий**: JavaScript является мощным инструментом для обработки событий на веб-страницах. Он может реагировать на действия пользователей, такие как клики мышью, нажатия клавиш и другие.
* **Манипуляция DOM**: JavaScript может изменять структуру и содержимое веб-страницы, обеспечивая динамическое обновление контента без перезагрузки страницы.
* **Асинхронные операции**: JavaScript позволяет выполнять асинхронные операции, такие как загрузка данных с сервера, без блокирования выполнения других задач.
* **Серверный JavaScript**: С помощью платформы Node.js JavaScript может быть использован на серверной стороне для создания высокопроизводительных и масштабируемых веб-приложений.
* **Множество фреймворков и библиотек**: JavaScript имеет обширное сообщество разработчиков и большое количество библиотек и фреймворков, таких как React, Angular, Vue.js, и другие, что упрощает создание сложных веб-приложений.
* **Кроссплатформенность**: JavaScript может быть использован для разработки веб-приложений на различных платформах, включая веб, мобильные устройства и даже десктоп.
* **Открытый стандарт**: JavaScript базируется на открытых стандартах ECMAScript, что обеспечивает его интероперабельность и расширяемость.

Веб-страницы частично обрабатываются с помощью JavaScript на компьютере пользователя. Это снижает нагрузку на сервер: часть операций выполняется без запросов к нему, что экономит время и трафик.

**2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

**2.1 Описание предметной области**

Предметная область - область теории, рассматривасмая в пределах отдельного рассуждения, научной теории. Также под предметной областью принято понимать часть реального мира, подлежащего изучению для организации управления и ,в конечном счете, автоматизации.

Предметной областью данной курсовой работы, является онлайн информационная система для продуктовых магазинов.

Архитектура сайта – это систематизация информации и навигации по

ней с целью помочь потенциальным заказчикам находить нужные им данные.

Хорошо продуманная архитектура сайта обеспечивает меньшее время на поиск запрашиваемой информации и гарантирует популярность среди пользователей благодаря простоте использования сайта.

При разработке архитектуры сайта необходимо учитывать то, как

представление информации повиляет на продвижение товаров и услуг на интернет-рынке. В процессе создания структуры нового сайта, либо

оптимизации структуры уже существующего, необходимо представлять

информацию таким образом, чтобы повысить посещаемость сайта и привлечь

внимание потенциальных покупателей к наиболее важным его разделам, исходя их предлагаемых товаров и услуг.

Важным элементом в разработке интернет-магазина является интерфейс, так как с помощью него будет происходить взаимодействие с клиентом.

Благодаря грамотному расположению блоков на странице клиент сможет найти интересующую информацию о товарах или же предоставляемых услугах компании.

Для разработки были использованы следующие технологии:

* HTML – этот язык является базовым в области технологий создания сайтов, так как относительно легок в освоении.
* CSS – формальный язык, преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML.

**2.2** **Создание окружения для проекта**

В качестве редактора кода будет использован XCode. Создаем папку в нем, которую назовем “ kursovaya”, в нем и будут все файлы проекта.

​Сайт будет состоять лишь из трёх страницы. Создадим для нее файл file.html. После этого создаем css файл.

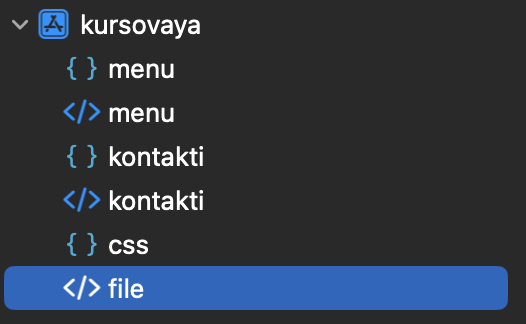
​Окружение готово и теперь мы можем приступить к самой разработке (рис. 1)

Рис. 1. Основной файл

**2.3 Верстка HTML страницы**

Открываем ранее созданный “file.html” и пишем стандартный HTML5 код (рис. 2).

Рис. 2

Элемент <!DOCTYPE> предназначен для указания типа текущего документа — DTD (document type definition, описание типа документа). Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях.

Тег <html> является контейнером, который заключает в себе все содержимое веб-страницы, включая теги <head> и <body>. Открывающий и закрывающий теги <html> в документе необязательны, но хороший стиль диктует непременное их использование.

Элемент <head> содержит машиночитаемую информацию ([metadata](https://developer.mozilla.org/ru/docs/Glossary/Metadata)) о документе, например его [заголовок](https://developer.mozilla.org/ru/docs/Web/HTML/Element/title), [скрипты](https://developer.mozilla.org/ru/docs/Web/HTML/Element/script) и [страницы стилей](https://developer.mozilla.org/ru/docs/Web/HTML/Element/style).

Элемент<body> представляет собой контент (содержимое) документа HTML. В документе может быть только один элемент <body>.

Затем мы подключаем наши файлы через <link> в элементе <head> (рис. 3)

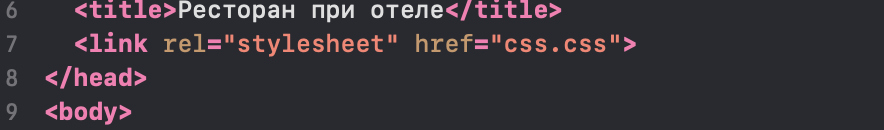


Рис. 3

Создаем <div>, куда приписываем название нашего ресторана.(рис.4)

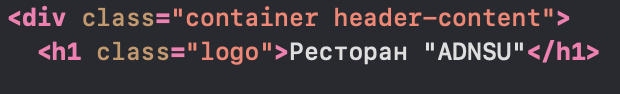


Рис. 4

Затем создаем меню через <ul> , <li>(рис.5)

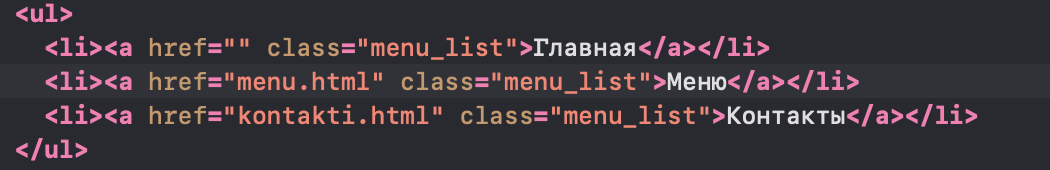


Рис. 5

HTML-элемент <ul> используется для неупорядоченного списка - в частности для маркированного списка.

HTML-элемент <li> используется для создания элементов списка. Он также должен находиться в родительском элементе: упорядоченном списке ([<ol>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/ol)), неупорядоченном списке ([<ul>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/ul)), или меню ([<menu>](https://developer.mozilla.org/ru/docs/Web/HTML/Element/menu)).

Создадим ещё один <div>, куда будем приписывать меню для выбора языка страницы. (Рис. 6)

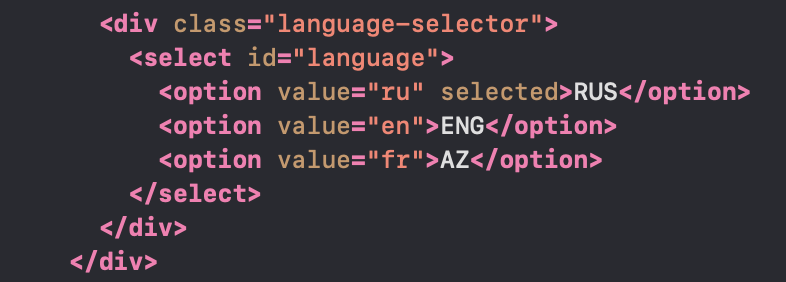


Рис. 6

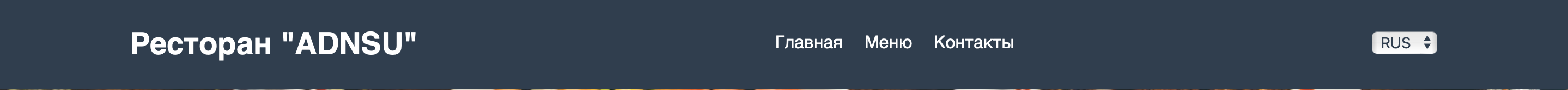
После задания css стилей, навбар примет такой вид. (Рис. 7)

Рис. 7

Теперь перейдём к разработке секции для перехода на страницу «Меню».

Сначала создаем элемент <section> и прописываем описание(class=”hero”): (Рис. 8,9)

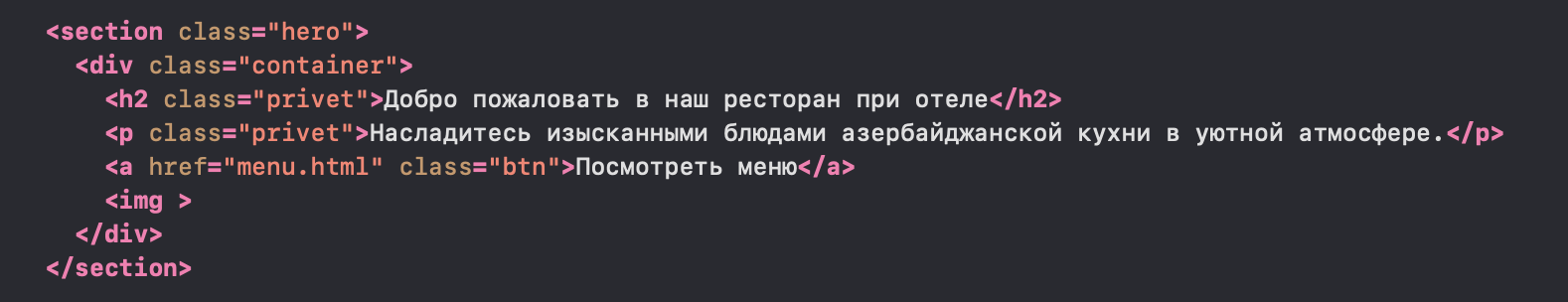


Рис. 8

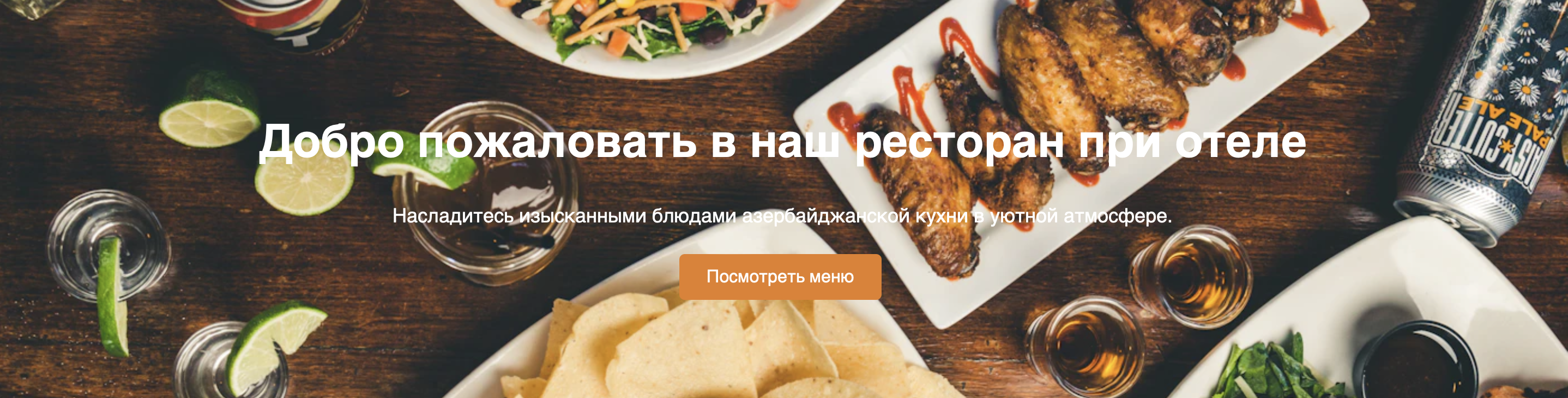


Рис. 9

HTML-элемент <a> определяет гиперссылку для перехода на определённое место на странице или на другую страницу в Интернете. Также он может быть использован (в устаревшем варианте) для создания якоря — это место назначения для гиперссылок внутри страницы: так ссылки не ограничены только в перемещении между страницами.

Теперь приступаем к последующей секции для резервирования столика.(Рис. 10)

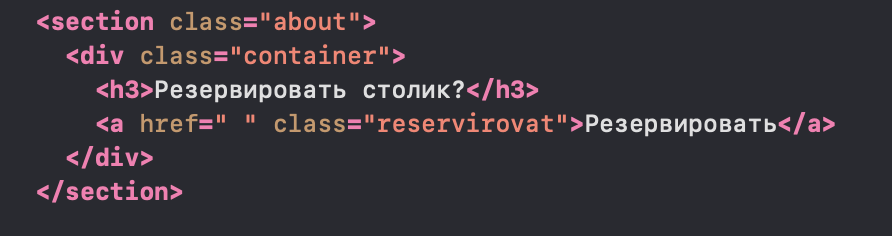


Рис. 10

И последняя секция- это секция <footer>.

**Footer (футер, подвал)** — блок в нижней части страницы. Содержит полезную, но не первостепенную информацию. Виден на всех страницах сайта. В футер можно вынести: копирайт, название студии, которая разрабатывала сайт, контакты. Иногда в футере дублируются пункты меню. (Рис. 11)



Рис. 11