Название учебного заведения: Geekbrains

Специальность: «Быстрый старт. Frontend-программист. Цифровые профессии»

Тема дипломной работы: Разработка веб-сайта для мебельной компании..

ФИО студента: Лошакова Лилия Юрьевна

Ссылка на GitHub: https://github.com/Riri546?tab=overview&from=2023-08-01&to=2023-08-11

​

Web версия дипломной работы находиться по адресу

​

Место и год написания - Краснодар, 2023

Тема: Разработка веб-сайта для мебельной компании.

Цель: Изучить особенности разработки веб-сайта на примере мебельной компании.

Задачи:

1. Изучить литературу, касающуюся темы исследования;
2. Рассмотреть основные виды и методов добавления эффектов наведения и нажатия;
3. Ознакомиться с семантическими элементами;
4. Создать в Figma дизайн веб-сайта;
5. Разработать сайт в соответствии с макетом;
6. Добавить адептов сайта и различного рода эффекты.

Инструменты: Figma, JS, HTML, CSS, SASS, Валидатор, Типограф.

Структура дипломного проекта:

Оглавление

Введение (2 - 3 стр)

Глава 1 Основы создания веб-сайта (~ 15 стр)

1.1 Что такое веб-сайт и для чего он нужен

1.2 Основные инструменты для создания сайта

1.3 Виды и методы добавления эффектов наведения и нажатия

1.4 Что такое семантические элементы

Глава 2 Подготовка к написанию сайта (~ 15 стр)

2.1 Создание макета в Figma

2.2 Разработка контентной части

2.3 Разработка изменений сайта при нажатии и наведении

Глава 3 Разработка веб-сайта (~ 20 стр)

3.1 Создание шапки сайта

3.2 Создание контентной части

3.3 Разработка подвала сайта

3.4 Добавление адаптива

3.5 Добавление эффектов наведения и нажатия

3.6 Проверка внешнего вида на разных разрешениях экрана

3.7 Проверка на валидность

3.8 Проверка типографики

Заключение (~ 4стр)

Список используемой литературы

Приложения

Оглавление

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Введение | ………………………………………………………………………….. | | | | | | | | | | | | | | | | |  |
| Глава 1 Основы создания веб-сайта | | | | | | | | ……………………………………………... | | | | | | | | | |  |
| 1.1 Что такое веб-сайт и для чего он нужен | | | | | | | | | | | | | ………………………………. | | | | |  |
| 1.2 Основные инструменты для создания сайта | | | | | | | | | | | | | | …………………………. | | | |  |
| 1.3 Виды и методы добавления эффектов наведения и нажатия | | | | | | | | | | | | | | | | | ………. |  |
| 1.4 Что такое семантические элементы | | | | | | | | | | | | …………………………………... | | | | | |  |
| Глава 2 Подготовка к написанию сайта | | | | | | | | | | | ………………………………………... | | | | | | |  |
| 2.1 Создание макета в Figma | | | | | | | | …………………………………………....... | | | | | | | | | |  |
| 2.2 Разработка контентной части | | | | | | | | | | …………………………………………. | | | | | | | |  |
| 2.3 Разработка изменений сайта при нажатии и наведении | | | | | | | | | | | | | | | ……………... | | |  |
| Глава 3 Разработка веб-сайта | | | | ………………………………………………....... | | | | | | | | | | | | | |  |
| 3.1 Создание шапки сайта | | | | | | ………………………………………………... | | | | | | | | | | | |  |
| 3.2 Создание контентной части | | | | | | | | | …………………………………………... | | | | | | | | |  |
| 3.3 Разработка подвала сайта | | | | | | | | …………………………………………….. | | | | | | | | | |  |
| 3.4 Добавление адаптива | | | | | ………………………………………................. | | | | | | | | | | | | |  |
| 3.5 Добавление эффектов наведения и нажатия | | | | | | | | | | | | | | …………………………. | | | |  |
| 3.6 Проверка внешнего вида на разных разрешениях экрана | | | | | | | | | | | | | | | | …………... | |  |
| 3.7 Просверкана валидность | | | | | | | | ……………………………………………... | | | | | | | | | |  |
| 3.8 Проверка типографики | | | | | | | ………………………………………………. | | | | | | | | | | |  |
| Заключение | | ……………………………………………………………………….. | | | | | | | | | | | | | | | |  |
| Список используемой литературы | | | | | | | | ……………………………………………... | | | | | | | | | |  |
| Приложения | | | ……………………………………………………………………… | | | | | | | | | | | | | | |  |

Введение

Первый в мире сайт info.cern.ch появился 6 августа 1991 года. Его создатель, Тим Бернерс-Ли, опубликовал на нём описание новой технологии World Wide Web, основанной на протоколе передачи данных HTTP, системе адресации URI и языке гипертекстовой разметки HTML. На сайте были описаны принципы установки и работы серверов и браузеров. Сайт стал и первым в мире интернет-каталогом, так как позже Тим Бернерс-Ли разместил на нём список ссылок на другие сайты.

Со временем технологии улучались и на данный момент у веб-разработчиков есть возможность непросто размещать текст на странице сайта и прикреплять к нему ссылки, появились такие возможности, как добавление стилизации, позиционирование, трансформирование, анимированные и так далее.

На сегодняшний день веб-сайты играют важную роль в жизни каждого человека. С помощью сайтов мы можем записаться к врачу, купить одежду, узнать о новых компаниях или найти полезную и интересную информацию.

Актуальность данной темы обусловлена тем, что в наше время каждая компании нуждается в узнаваемости, при этом веб-сайт является главным инструментом для продвижения бизнеса, так как на нем может быть размещена информация о компании, список товаров и услуг, форма подписки на рассылку с новыми предложениями, возможность покупки в онлайн режиме, не выходя из дома, а также возможность заказать звонок. Вся эта информация, размещённая в удобном для пользователя формате, позволяет клиенту, не выходя из дома ознакомиться с компанией и принять решении о покупке товаров или услуг.

В соответствии с выбранной темой, разберем разработку сайта для мебельной компании, так как данный бизнес покрывает не маловажную часть потребностей клиентов. При этом многие клиенты не знают, где можно заказать качественную мебель, подходящую под их размеры, вид деятельности и желаемый внешний вид.

Цель: Изучить особенности разработки веб-сайта на примере мебельной компании.

Задачи:

1. Изучить литературу, касающуюся темы исследования;
2. Рассмотреть основные виды и методов добавления эффектов наведения и нажатия;
3. Ознакомиться с семантическими элементами;
4. Создать в Figma дизайн веб-сайта;
5. Разработать сайт в соответствии с макетом;
6. Добавить адептов сайта и различного рода эффекты.

Инструменты: Figma, JS, HTML, CSS, SASS, Валидатор, Типограф.

Глава 1 Основы создания веб-сайта

1.1 Что такое веб-сайт и для чего он нужен

Веб-сайт - это интерактивная веб-страница или набор страниц, предоставляющих пользователю доступ к информации, изображениям, видео и другим мультимедийным элементам через интернет. Он может быть создан с различными целями, включая продажу товаров и услуг, предоставление информации о компании и ее продуктах, общение с пользователями, обучение, развлечение и т. д.

Дизайн и функционал сайта могут быть разнообразными, включая такие элементы, как интерактивность, анимация, аудио и видео контент, формы обратной связи, панели управления, базы данных, онлайн-чаты и другие функции. Сайты могут быть статическими, где контент фиксирован, или динамическими, где контент генерируется и изменяется в зависимости от действий пользователя.

Для создания сайтов используются разнообразные технологии, языки программирования и инструменты, такие как HTML, CSS, JavaScript, PHP, Python, Ruby, CMS (системы управления контентом), фреймворки и другие. Кроме того, важно уделять внимание аспектам безопасности, скорости загрузки страниц и оптимизации для поисковых систем.

Сегодня сайт является неотъемлемой частью современного бизнеса и представляет собой мощный инструмент для привлечения клиентов и продвижения бренда. Он может быть использован как малыми предприятиями, так и крупными компаниями, и становится все более доступным благодаря развитию технологий и услуг веб-разработки.

Веб-сайт состоит из множества элементов, каждый из которых выполняет уникальную функцию и направлен на улучшение пользовательского опыта. Давайте рассмотрим некоторые из основных элементов:

1. Шапка (Header) – это верхняя часть сайта, которая содержит логотип компании, основное меню, контактную информацию и другие важные элементы навигации. Она является визитной карточкой сайта и помогает пользователям быстро ориентироваться.
2. Навигация (Navigation) – это элемент, который облегчает поиск нужной информации на сайте. Он может быть представлен в виде меню, ссылок, выпадающих списков и других интерактивных элементов. Хорошая навигация помогает пользователям быстро и легко перемещаться по сайту, что повышает удобство пользования.
3. Содержание (Content) – это самая важная часть сайта, которая должна быть структурирована и информативна для посетителей. Хорошо организованное содержание помогает пользователям быстро находить нужную информацию и улучшает их взаимодействие с сайтом. Важно, чтобы контент был понятным, легкочитаемым и содержал полезную информацию для посетителей.
4. Боковая панель (Sidebar) – это дополнительный блок на сайте, расположенный по бокам страницы, который содержит различные виджеты и информацию. В боковой панели можно размещать рекламу, теги, популярные посты и многое другое.
5. Футер (Footer) – это нижняя часть сайта, которая содержит дополнительную информацию о сайте. В футере обычно размещаются ссылки на социальные сети, контактную информацию, копирайт и другие полезные элементы.
6. Кнопки (Buttons) – это элементы интерфейса, которые помогают пользователям взаимодействовать с сайтом. Кнопки могут выполнять различные функции, например, они могут быть использованы для совершения покупки, заказа звонка или подписки на рассылку.
7. Формы (Forms) – это элементы, которые позволяют пользователям отправлять информацию на сайт. Формы могут использоваться для обратной связи, регистрации пользователя, оформления заказа и других целей. Они позволяют пользователям удобно и безопасно передавать нужные данные.
8. Изображения и видео (Images and Video) – это важные элементы, которые помогают визуализировать информацию на сайте и делают его более привлекательным для посетителей. Они могут быть использованы для улучшения пользовательского опыта и привлечения внимания к ключевым сообщениям.
9. Социальные кнопки (Social Media Buttons) – это инструменты, которые помогают пользователям легко делиться контентом сайта в социальных сетях. Они увеличивают его видимость и способствуют распространению информации, что может привести к привлечению новых посетителей и повышению уровня вовлеченности.
10. Карта сайта (Sitemap) – это страница, которая содержит структуру сайта и помогает поисковым системам более эффективно индексировать его содержимое. Это существенно для улучшения видимости и доступности сайта, так как позволяет посетителям быстро найти нужную им информацию и лучше понять его структуру.

Каждый из этих элементов имеет свою уникальную функцию и может быть использован для улучшения пользовательского опыта на сайте. Эффективное сочетание всех этих элементов поможет сделать сайт более привлекательным и удобным для посетителей.

На сегодняшний день существует множество разных типов веб-сайтов, каждый из которых обладает своими уникальными особенностями и функционалом. Поэтому знакомство с различными типами сайтов может помочь вам сделать правильный выбор и найти подходящий тип сайта для решения вашей конкретной задачи.

Один из самых распространенных типов сайтов - это сайты-визитки. Они часто используются для представления компании, ее товаров или услуг. Обычно на таких сайтах содержится основная информация о компании, ее контактные данные и некоторые изображения, позволяющие представить визуальное представление о компании.

Еще один тип сайтов - это корпоративные сайты. Они предоставляют детальную информацию о компании, ее продуктах, услугах и контактных данных. Кроме того, на таких сайтах может быть раздел для новостей и пресс-релизов, а также документация для скачивания.

Однако это только два примера из множества возможных типов сайтов. В зависимости от ваших потребностей, вы можете также рассмотреть другие типы, такие как интернет-магазины, блоги, форумы и многое другое. Важно определить свои цели и требования, чтобы выбрать тип сайта, который наилучшим образом соответствует вашим потребностям.

Интернет-магазины предоставляют возможность пользователям приобретать товары и услуги онлайн. Они обычно оснащены удобными каталогами продуктов, функцией добавления товаров в корзину и системой оплаты, что делает процесс покупки максимально удобным и безопасным.

Блоги являются площадкой, где авторы публикуют свои мнения, статьи, рассказы и другой интересный контент. Они часто содержат категории, тэги, комментарии и форму для обратной связи, что позволяет читателям активно обсуждать и выражать свое мнение по поводу опубликованного материала.

Форумы представляют собой площадку для общения пользователей, где они могут создавать темы, вступать в дискуссии и обмениваться мнениями. Форумы обычно имеют разделы для различных тем и подразделы для более углубленного обсуждения конкретных вопросов. Это идеальное место для обмена опытом, получения советов и нахождения ответов на интересующие вопросы.

Социальные сети являются платформами, где пользователи могут создавать свои профили, общаться друг с другом, делиться фотографиями и видео, играть в игры и многое другое. Кроме того, эти площадки также предоставляют возможности для рекламы и продвижения бренда.

Сайты-каталоги, в свою очередь, являются местами, где собраны ссылки на различные площадки, разделенные по тематикам. Они могут быть как универсальными, так и специализированными.

Кроме вышеперечисленных типов платформ, существуют также порталы, агрегаторы новостей, онлайн-сервисы и другие. Каждый из них обладает своими уникальными особенностями и возможностями, поэтому важно выбрать наиболее подходящий вариант, отвечающий вашим потребностям.

1.2 Основные инструменты для создания сайта

Инструменты веб-разработки существенно облегчают процесс создания веб-сайтов и значительно расширяют их функциональные возможности. Эти инструменты можно разделить на несколько категорий:

* Компьютер и операционная система - необходимые инструменты, которые понятны и очевидны.
* Инструменты для работы с текстом - включают текстовые редакторы, которые используются для написания программного кода.
* Веб-браузеры - это программы, которые позволяют пользователям просматривать и отображать веб-страницы. Если разработчик хочет, чтобы его веб-сайт был доступен для пользователей разных браузеров, то необходимо протестировать его в разных веб-браузерах.
* Отладчик браузера - это инструмент, который может быть использован в качестве черновика при написании кода. Внося изменения в отладчик, можно быть уверенным, что веб-сайт не сломается. Если результат не устраивает, достаточно просто обновить страницу. Кроме того, отладчик помогает обнаружить ошибки в коде, возможные проблемы со стилями и так далее.

В процессе разработки веб-проектов часто используются различные инструменты и программы. Некоторые из них включают графические редакторы, которые используются для проектирования шаблонов, верстки и графического наполнения. Они помогают создавать красивый и профессиональный дизайн веб-страниц.

Другой важный инструмент - система контроля версий, которая позволяет хранить разные состояния проекта и облегчает совместную работу. В системе контроля отслеживаются все изменения, вносимые в проект, и таким образом программисты могут легко делиться кодом, поддерживать его и минимизировать конфликты при совместной работе.

FTP клиент - это программа для передачи файлов на сервер (например, веб-сервер), их загрузки и редактирования. Он очень полезен при разработке веб-сайтов, так как позволяет легко обмениваться файлами между локальной машиной и сервером.

Наконец, фреймворки - это удобные инструменты, предлагающие готовые решения для различных задач. Они содержат наборы инструментов, библиотек и стандартных практик, которые значительно упрощают и ускоряют разработку веб-проектов. Фреймворки предлагают структуру и организацию проекта, что позволяет разработчикам сосредоточиться на реализации функциональности, а не на создании всего с нуля.

1.3 Виды и методы добавления эффектов наведения и нажатия

Псевдоклассы предоставляют возможность добавить особые классы к элементам. Они применяются к объектам, которые либо отсутствуют в структуре веб-страницы, либо не могут быть выбраны с помощью обычных селекторов. Например, псевдокласс можно использовать для стилизации первой буквы или первой строки в абзаце.

Вы, наверняка, замечали на различных сайтах, что при наведении курсора мыши на пункт меню происходят различные изменения. Фон меняет свой цвет, ссылки становятся другого цвета, а также изменяются шрифт и его размер. Все это возможно благодаря применению псевдоклассов. Давайте рассмотрим их синтаксис (см. Рис. 1).



Рис. 1 Синтаксис псевдокласса

После селектора следует двоеточие, а затем указывается название псевдокласса без пробелов. Наиболее часто используемые псевдоклассы определяют состояние элемента и позволяют изменить его визуальное представление.

Ниже представлены несколько примеров популярных псевдоклассов:

1. a:link — применяется к ссылке, которая еще не была посещена пользователем.
2. a:visited — применяется к ссылке, которая уже была посещена пользователем.
3. a:hover — применяется к любому элементу, над которым находится курсор мыши.
4. a:focus — применяется к любому элементу, на котором установлен фокус с помощью мыши или клавиатуры.

Использование этих псевдоклассов позволяет создавать интерактивные и улучшенные пользовательские интерфейсы.

5. Псевдоэлемент "a:active" ссылается на активированный пользователем элемент. Псевдоэлементы позволяют ввести несуществующие элементы в веб-документ и придать им определенные стили. Они были введены в CSS1, но стали доступны только в CSS2.1. В начале использовалось одно двоеточие, но в CSS3 используется двойное двоеточие для отличия от псевдоклассов. Большинство современных браузеров поддерживают оба типа синтаксиса псевдоэлементов, за исключением Internet Explorer 8, который поддерживает только одно двоеточие. Поэтому рекомендуется использовать его для повышения совместимости. C помощью свойства "content" можно изменить внешний вид части элемента.

В этом разделе мы рассмотрим некоторые часто используемые псевдоэлементы в CSS:

1. :first-letter — этот псевдоэлемент позволяет выбирать первую букву каждого абзаца. Обратите внимание, что он применяется только к блочным элементам.
2. :first-line — этот псевдоэлемент выбирает первую строку текста внутри элемента. Он также применяется только к блочным элементам.
3. :before — этот псевдоэлемент вставляет генерируемое содержимое перед указанным элементом.
4. :after — этот псевдоэлемент добавляет генерируемое содержимое после указанного элемента.

Таким образом, эти псевдоэлементы могут быть полезными инструментами для стилизации и улучшения внешнего вида вашего веб-сайта.

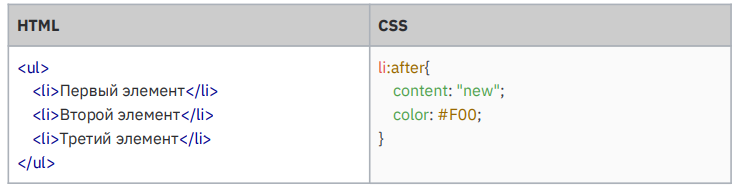


Рис. 2 Пример использования псевдоэлементов

После всех элементов списка li, изображенных на рисунке 2, появится текст "new" красного цвета. Если добавить плавные переходы для элементов, то для общего свойства transition будут использованы его составляющие, аналогично свойствам background, background-color, background-image и т.д. Давайте рассмотрим, из чего состоит общее свойство transition:

1. Свойство transition-property. Оно содержит название CSS-свойств, к которым будет применен эффект перехода. Это свойство не наследуется.
   1. Значение
      1. none - отсутствие свойства для перехода.
      2. all - значение по умолчанию. Это применяет эффект перехода ко всем свойствам элемента.
      3. свойство - определяет список css-свойств, перечисленных через запятую, которые участвуют в переходе.
2. Продолжительность перехода transition-duration задает промежуток времени, в течение которого должен осуществляться переход. Это свойство не наследуется.
   1. Значения
      1. Время перехода указывается в секундах или миллисекундах, например, 1s или 5ms, или 0.3s.
3. Функция перехода transition-timing-function. Это свойство задает временную функцию, которая определяет скорость перехода объекта от одного значения к другому. Если вы определяете более одного перехода для элемента, например, цвет фона элемента и его положение, вы
   1. Значения
      1. ease — это функция по умолчанию, которая обеспечивает плавный переход с медленным началом, быстрым ускорением и замедлением в конце. Она соответствует кривой cubic-bezier(0.25, 0.1, 0.25, 1).
      2. linear — переход происходит равномерно на протяжении всего времени, без колебаний в скорости. Он соответствует кривой cubic-bezier(0, 0, 1, 1).
      3. ease-in — переход начинается медленно и плавно ускоряется в конце. Он соответствует кривой cubic-bezier(0.42, 0, 1, 1).
      4. ease-out — переход начинается быстро и плавно замедляется в конце. Он соответствует кривой cubic-bezier(0, 0, 0.58, 1).
      5. ease-in-out — переход медленно начинается и медленно заканчивается. Этот стиль соответствует cubic-bezier(0.42,0,0.58,1).
      6. cubic-bezier(x1, y1, x2, y2) — позволяет вручную устанавливать значения от 0 до 1 для кривой ускорения.
4. Задержка перехода transition-delay. Это необязательное свойство позволяет изменять свойства с некоторой задержкой, а не моментально. Оно не наследуется.
   1. Время задержки перехода указывается в секундах или миллисекундах.
5. Краткая запись перехода transition. Все свойства, отвечающие за изменение внешнего вида элемента, можно объединить в одно свойство transition.

Для создания анимации в CSS3 используется свойство @keyframes. Это свойство представлено на рисунке 3 и является контейнером для различных свойств оформления.

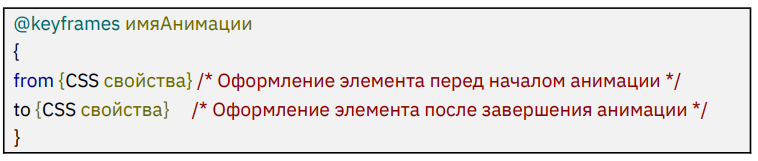


Рис.3 Пример записи свойства @keyframes

Для добавления анимации к элементу необходимо использовать CSS3 свойство "animation". В нем нужно указать имя анимации (1 значение) и время, в течение которого она будет выполняться (2 значение). Также можно задать количество повторов анимации (3 значение) (см. рис. 4).



Рис.4 Пример записи анимации

Длительность анимации animation-duration определяет, сколько времени занимает проигрывание анимации. Это свойство не наследуется от родительских элементов. Если не задано явно, по умолчанию устанавливается значение 0. (См. таблицу 1).

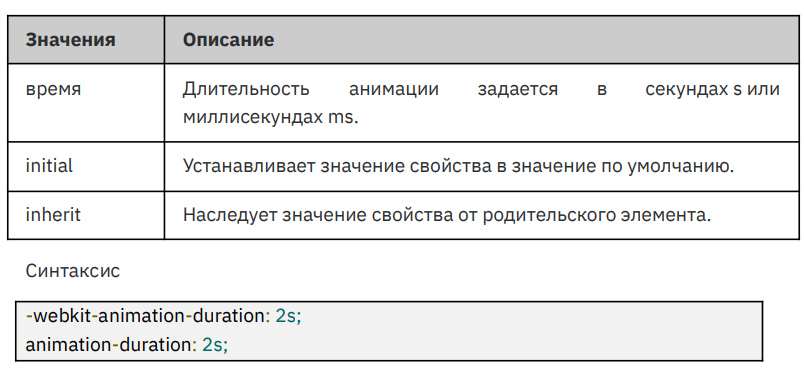


Табл.1 Значения свойств animation-duration

1.4 Что такое семантические элементы

Семантические HTML-элементы — это теги, которые служат для описания содержимого, находящегося внутри них. Они предоставляют информацию о структуре и значимости веб-страницы, что помогает поисковым системам лучше понимать ее содержание. Использование семантических HTML-элементов также сделает ваш код более понятным и легко поддающимся обслуживанию, так как каждый элемент будет иметь ясное назначение. В HTML5 появились новые семантические элементы, которые позволяют лучше структурировать веб-страницы и давать содержимому смысловую значимость. Отличительной особенностью всех семантических элементов является то, что они сами по себе ничего не делают. Например, элемент <video> используется для вставки полноценного видеоплеера на веб-страницу.

В данном случае возникает вопрос о том, стоит ли использовать новые элементы. Рассмотрим несколько причин, почему это целесообразно:

* + - 1. Более удобное редактирование и сопровождение. Используя новые элементы, можно с легкостью редактировать и обновлять веб-страницы, что облегчает работу разработчиков.

1. Оптимизация поисковых движков. Современные поисковые роботы уже проверяют наличие некоторых семантических элементов HTML5, чтобы получить больше информации о веб-страницах, которые они индексируют. Это может повысить видимость и рейтинг вашего сайта в поисковой выдаче.
2. Поддержка будущих возможностей. Новые браузеры и инструменты для редактирования веб-страниц используют всю мощь семантических элементов. Использование их сейчас гарантирует, что ваш сайт будет готов к будущим функциям и возможностям.

Внедрение новых элементов HTML5 не только улучшает функциональность и эффективность вашего сайта, но и подготавливает его для будущих инноваций в веб-разработке.

Рассмотрим назначение новых элементов более подробно:

Семантические элементы для работы со структурой страницы. Один из таких элементов - <header>. Он группирует вводные и навигационные элементы, хотя не является обязательным. В <header> можно разместить заголовки, заключить содержание раздела страницы, создать форму поиска или вставить логотипы. В html-документе может быть несколько элементов <header>, и они могут находиться в любой части страницы.



Рис. 5 Синтаксис семантического элемента <header>

Элемент <nav> предназначен для создания навигационных блоков на веб-странице или веб-сайте и может быть размещен как внутри <header>, так и в других частях страницы. Важно отметить, что на одной странице может быть несколько элементов <nav>. Этот элемент не заменяет теги <ul> или <ol>, а просто обрамляет их. Также не каждая группа ссылок на странице должна быть обернута в элемент <nav>. Он предназначен в первую очередь для разделов, содержащих основные блоки навигации.



Рис. 6 Синтаксис семантического элемента <nav>

В панели навигации можно использовать разнообразные элементы, а не ограничиваться только списками (см. Рис. 7):



Рис. 7 Пример использования элемента <nav>

Элемент <footer> представляет нижний колонтитул страницы или статьи. Обычно он содержит информацию об авторе, копирайте и дополнительные данные. Если <footer> используется для всей страницы, в нем также может присутствовать информация о авторских правах, условиях использования, контактной информации и ссылках на связанное содержимое. В одном веб-документе может быть несколько элементов <footer>. Интересно, что как каждая страница, так и каждая статья может иметь свой собственный <footer>, и его также можно поместить внутрь элемента <blockquote>, чтобы указать источник цитирования.



Рис. 8 Синтаксис семантического элемента <footer>

Элемент <article> используется для группировки различных записей, таких как публикации, статьи, записи блога или комментарии. Он представляет собой самостоятельный блок, который может быть использован несколько раз и обычно начинается с заголовка. Элемент <article> может также содержать другие вложенные элементы <article>, которые имеют схожее содержание с внешней статьей. Если на странице присутствует только одна статья, состоящая из заголовка и текстового содержания, то нет необходимости оборачивать ее в элемент <article>.

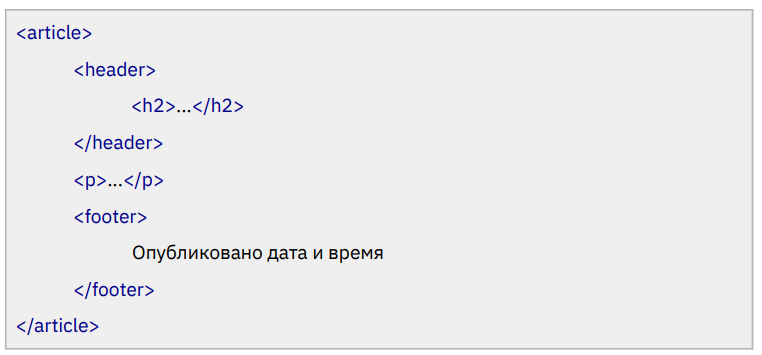


Рис. 9 Синтаксис семантического элемента <article>

Элемент <section> представляет собой многофункциональный компонент документа, который группирует тематически связанное содержимое и обычно содержит заголовок. В отличие от элемента <div>, <section> не используется для обертывания блоков. Он может содержать различные типы информации, такие как оглавления, разделы научных публикаций и программы мероприятий. На домашней странице сайта также можно использовать <section> для разделения вводной информации, новостей и контактов.



Рис. 10 Синтаксис семантического элемента <section>

Элемент <aside> предназначен для группировки содержимого, которое связано с основным контентом, но может быть рассмотрено как отдельный блок. Удаление этого элемента не должно повлиять на понимание основного содержания страницы. Обычно <aside> используется в качестве боковой колонки, например, в книгах, и может содержать такие элементы, как навигация, цифровые данные, цитаты, рекламные блоки или архивы. Однако, не следует использовать <aside> для простого позиционирования блоков в стороне.

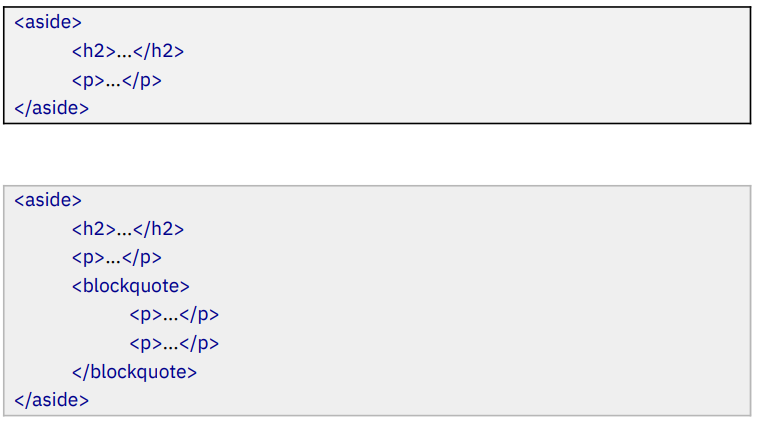


Рис. 11 Синтаксис семантического элемента <aside>

На рисунке 12 показан способ применения элемента <aside>:

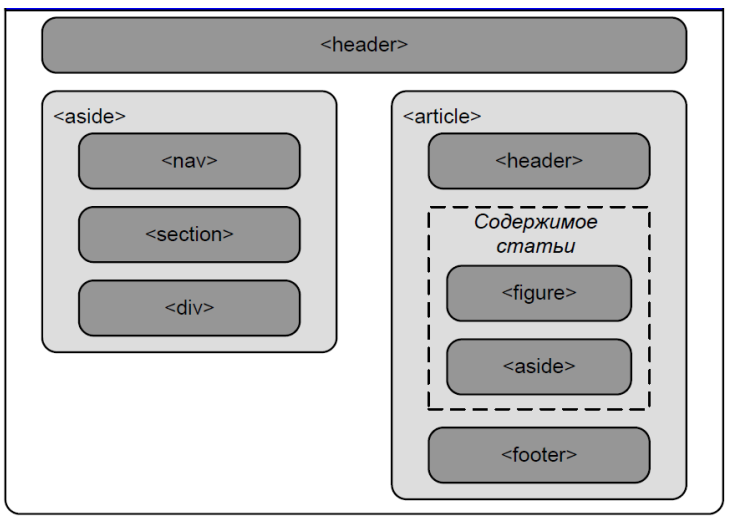


Рис. 12 Возможная структура элемента <aside>