Название учебного заведения — Geekbrains

Специальность: «Разработчик - Тестировщик» — 3291

Тема дипломной работы

Важность принципов Agile и DevOps в тестировании программного обеспечения на примере тестового сайта .

ФИО студента - Хижнякова Анна Викторовна

Ссылка на GitHub

​

Web версия дипломной работы находиться по адресу

​

Место и год написания - Москва, 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

Веб сайт – это интерактивная веб-страница или набор страниц, которые содержат информацию, изображения, видео и другие мультимедийные элементы, доступные через интернет. Он может быть создан для различных целей, таких как продажа товаров и услуг, информирование о компании и ее продуктах, общения с пользователями, обучения, развлечения и т.д.

Сайт может иметь различный дизайн и функционал, включая интерактивность, анимацию, аудио и видео контент, формы обратной связи, панели управления, базы данных, онлайн-чаты и другие функции. Сайты могут быть статическими, содержащими фиксированный контент, или динамическими, с генерируемым контентом, который меняется в зависимости от действий пользователя.

Для создания сайтов используются различные технологии, языки программирования и инструменты, такие как HTML, CSS, JavaScript, PHP, Python, Ruby, CMS (системы управления контентом), фреймворки и другие. Также важно учитывать аспекты безопасности, скорости загрузки страниц и оптимизации под поисковые системы.

Сегодня сайт – это неотъемлемая часть современного бизнеса и представляет собой мощный инструмент для привлечения клиентов и продвижения бренда. Он может быть использован как для малого бизнеса, так и для крупных компаний, и становится все более доступным благодаря развитию технологий и услуг веб-разработки.

Веб-сайт состоит из множества элементов, каждый из которых имеет свою уникальную функцию и призван улучшить пользовательский опыт. Рассмотрим некоторые из основных элементов:

1. Шапка (Header) – верхняя часть сайта, которая содержит логотип компании, основное меню, контактную информацию и другие важные элементы навигации.
2. Навигация (Navigation) – элемент, который облегчает поиск нужной информации на сайте. Он может быть представлен в виде меню, ссылок, выпадающих списков и т.д.
3. Содержание (Content) – это самое важное содержимое сайта, которое должно быть четко структурировано и информативно для посетителей.
4. Боковая панель (Sidebar) – дополнительный блок на сайте, который содержит различные виджеты, такие как реклама, теги, популярные посты и т.д.
5. Футер (Footer) – нижняя часть, которая содержит дополнительную информацию о сайте, такую как ссылки на социальные сети, контактную информацию, копирайт и т.д.
6. Кнопки (Buttons) – элементы, которые помогают пользователям взаимодействовать с сайтом. Например, кнопки "Купить", "Заказать звонок", "Подписаться" и т.д.
7. Формы (Forms) – элементы, которые позволяют пользователям отправлять информацию на сайт, например, формы обратной связи, формы регистрации, формы заказа и т.д.
8. Изображения и видео (Images and Video) – элементы, которые помогают визуально представить информацию на сайте и сделать его более привлекательным для посетителей.
9. Социальные кнопки (Social Media Buttons) – элементы, которые помогают пользователям поделиться контентом в социальных сетях и увеличить его видимость.
10. Карта сайта (Sitemap) – страница, которая содержит структуру и помогает поисковым системам индексировать его содержимое.

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

Существует множество типов сайтов, каждый из которых имеет свои уникальные особенности и функционал. Ознакомление с различными типами может помочь в выборе подходящего типа для вашей конкретной задачи.

Сайты-визитки – часто используются для представления компании, товаров или услуг. Они обычно содержат основную информацию о компании, контактные данные и некоторые изображения.

Корпоративные сайты – предоставляют подробную информацию о компании, ее продуктах, услугах и контактных данных. Они могут также включать разделы для новостей и пресс-релизов, а также документацию для скачивания.

Интернет-магазины – позволяют пользователям покупать товары и услуги онлайн. Они обычно включают каталог продуктов, корзину для покупок и систему оплаты.

Блоги – место где, авторы публикуют свои мнения, статьи, рассказы и другой контент. Блоги обычно содержат категории, тэги, комментарии и форму для обратной связи.

Форумы – место где, пользователи могут общаться друг с другом, создавать темы и участвовать в дискуссиях. Форумы обычно содержат разделы для различных тем и подразделы для обсуждения конкретных вопросов.

Социальные сети – сайты, на которых пользователи могут создавать свои профили, общаться друг с другом, публиковать фотографии и видео, играть в игры и т.д. Социальные сети также могут предоставлять возможности для рекламы и продвижения бренда.

Сайты-каталоги – место где, собраны ссылки на другие площадки, разделенные по тематикам. Они могут быть как общими, так и специализированными.

Кроме того, существуют также другие типы, такие как порталы, агрегаторы новостей, онлайн-сервисы и т.д. Каждый тип имеет свои уникальные особенности и возможности, поэтому важно выбрать, наиболее подходящий вариант под ваши потребности.

1.2 Основные инструменты для создания сайта

Инструменты веб-разработки существенно упрощают создание веб-сайтов, расширяют их функционал. Они делятся на несколько категорий:

* компьютер и ОС – очевидные, но необходимые инструменты;
* инструменты организации текста – текстовые редакторы, где пишут программный код;
* веб-браузеры – браузеры читают и выводят код пользователям. Если программист хочет, чтобы веб-сайт был доступен любому желающему, нужно протестировать его на всевозможных браузерах;
* отладчик в браузере – может использоваться как некий черновик при написании кода. Внося код в отладчик можно быть уверенным, что сайт не испортится, а если результат не устроит, то достаточно просто обновить страницу. Также с помощью отладчика можно найти ошибку в коде, из-за которой не применяется стиль и т.д.
* графические редакторы – используются для проектирования шаблонов, верстки, графического наполнения;
* система контроля версий – инструмент для хранения разных состояний проекта и для совместной работы. В системе контроля отслеживаются изменения, вносимые в проект. Программа помогает делиться кодом, поддерживать его, минимизировать конфликты с выкладкой;
* ftp клиент – это клиент для передачи файлов (например, на веб-сервер), их выгрузки, редактирования;
* фреймворки – инструменты, которые предлагают готовые решения под разные задачи.

1.3 Виды и методы добавления эффектов наведения и нажатия

С псевдоклассами добавляются особые классы к элементам. Выбираются объекты, которые:

* отсутствуют в структуре веб-страницы;
* нельзя выбрать с помощью обычных селекторов. Например, первая буква или первая строка одного абзаца.

Вы наверняка замечали на сайтах: когда наводите мышкой на конкретный пункт меню, он меняет свой вид. У него изменяется цвет фона, цвет ссылки, даже шрифт или его размер. Это происходит благодаря псевдоклассам. Рассмотрим их синтаксис (Рис. 1).



Рис. 1 Синтаксис псевдокласса

После селектора ставится двоеточие. Сразу после него без пробела указывается название псевдокласса. Начнём с самых часто используемых псевдоклассов, определяющие состояние, простыми словами мы меняем состояние элемента, к которому добавляем псевдокласс.

1. a:link — ссылается на посещенную ссылку.
2. a:visited — ссылается на уже посещенную ссылку.
3. a:hover — ссылается на любой элемент, по которому проводят курсором мыши.
4. a:focus — ссылается на любой элемент, над которым находится курсор мыши.
5. a:active — ссылается на активированный пользователем элемент.

Псевдоэлементы позволяют ввести несуществующие элементы в веб-документ и придать им определённые стили. Псевдоэлементы появились ещё в CSS1, но пошли в релиз только в CSS2.1. В самом начале в синтаксисе использовалось одно двоеточие, но в CSS3 используется двойное двоеточие для отличия от псевдоклассов. Современные браузеры умеют понимать оба типа синтаксиса псевдоэлементов, кроме Internet Explorer 8. Он воспринимает только одно двоеточие. Поэтому надёжнее использовать одно. С помощью свойства content можно изменить внешний вид части элемента.

Рассмотрим часто используемые псевдоэлементы:

1. :first-letter — выбирает первую букву каждого абзаца, применяется только к
2. блочным элементам.
3. :first-line — выбирает первую строку текста элемента, применяется только к
4. блочным элементам.
5. :before — вставляет генерируемое содержимое перед элементом.
6. :after — добавляет генерируемое содержимое после элемента.

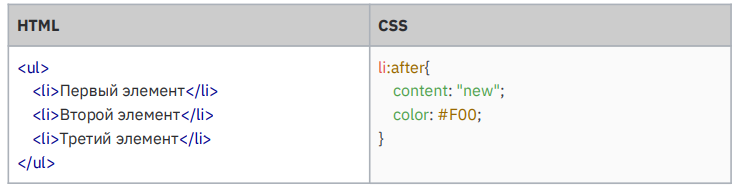


Рис. 2 Пример использования псевдоэлементов

После всех элементов списка li представленных на рисунке 2, появится текст new красного цвета.

При добавлении плавности переходов для элементов, для общего свойств transition существуют его составляющие (аналогично свойству background, background-color, background-image и т.д)

Рассмотрим из чего состоит общее свойство transition

1. Свойство transition-property. Содержит название CSS-свойств, к которым будет применен эффект перехода. Не наследуется.
   1. Значения
      1. none - отсутствие свойства для перехода.
      2. all - значение по умолчанию. Применяет эффект перехода ко всем свойствам элемента.
      3. свойство - определяет список css-свойств, перечисленных через запятую, участвующих в переходе.
2. Продолжительность перехода transition-duration задает промежуток времени, в течение которого должен осуществляться переход. 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. Свойство задаёт 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. iease — функция по умолчанию, переход начинается медленно, разгоняется быстро и замедляется в конце. Соответствует cubic-bezier(0.25,0.1,0.25,1).
      2. linear — переход происходит равномерно на протяжении всего времени, без колебаний в скорости. Соответствует cubic-bezier(0,0,1,1).
      3. ease-in — переход начинается медленно, а затем плавно ускоряется в конце. Соответствует cubic-bezier(0.42,0,1,1).
      4. ease-out — переход начинается быстро и плавно замедляется в конце. Соответствует cubic-bezier(0,0,0.58,1).
      5. ease-in-out — переход медленно начинается и медленно заканчивается. Соответствует cubic-bezier(0.42,0,0.58,1).
      6. cubic-bezier(x1, y1, x2, y2) — позволяет вручную установить значения от 0 до 1 для кривой ускорения.
4. Задержка перехода transition-delay. Необязательное свойство, позволяет сделать так, чтобы изменение свойства происходило не моментально, а с некоторой задержкой. 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. Данное свойство представленно на рисунке 3 и представляет собой контейнер, в который должны помещаться различные свойства оформления.

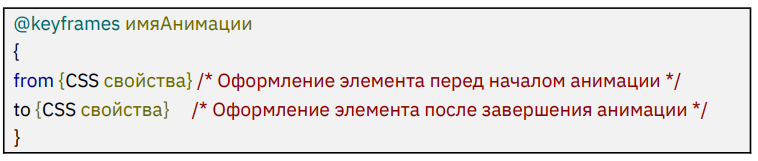


Рис.3 Пример записи свойства @keyframes

После того, как анимация была создана, необходимо добавить к элементу, который Вы хотите анимировать, CSS3 свойство animation и указать в нём имя анимации (1значение) и время (2 значение), в течение которого она будет выполняться.

Также Вы можете устанавливать количество повторов анимации (3 значение) (см. рис. 4).



Рис.4 Пример записи анимации

Длительность анимации animation-duration. Свойство устанавливает длительность анимации. Не наследуется. Значение по умолчанию 0 (Табл. 1).

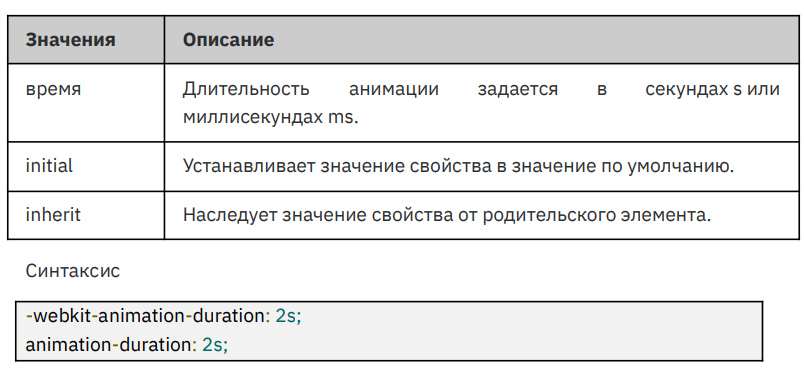


Табл.1 Значения свойств animation-duration

1.4 Что такое семантические элементы

Семантические HTML-элементы — это теги, которые описывают содержащийся в них контент. Они предоставляют информацию о контенте, которая может помочь поисковым системам понять структуру и значение веб-страницы.

Использование семантических HTML-элементов также делает код более читаемым и поддерживаемым, поскольку в нем четко указано назначение каждого элемента.

Новые семантические элементы HTML5 позволяют улучшить структуру веб-страницы, добавляя смысловое значение заключённому в них содержимому.

Все семантические элементы имеют отличительную особенность: они по существу ничего не делают. В противоположность, элемент <video>, например, вставляет в веб-страницу полноценный видеоплеер.

В данном случае возникает вопрос о целесообразности использования новых элементов.

Рассмотрим некоторые причины:

1. Более удобное редактирование и сопровождение;
2. Оптимизация поисковых движков. В настоящее время поисковые роботы уже проверяют на наличие некоторых семантических элементов HTML5, чтобы собрать всю возможную информацию об индексирующих их веб-страницах.
3. Поддержка будущих возможностей. В новых браузерах и инструментах редактирования веб-страниц будет использоваться весь диапазон предоставляемых семантическими элементами возможностей.

Рассмотрим назначение новых элементов более подробно:

Семантические элементы для работы со структурой страниц. Секционные элементы:

* + Элемент <header>

Группирует вводные и навигационные элементы, не является обязательным. Может содержать заголовки, оборачивать содержание раздела страницы, форму поиска или логотипы. В html-документе может содержаться одновременно несколько элементов <header>, и они могут располагаться в любой части страницы.



Рис. 5 Синтаксис семантического элемента <header>

* Элемент <nav>

Предназначен для создания блока навигации веб-страницы или всего веб-сайта, при этом не обязательно должен находиться внутри <header>. На странице может быть несколько элементов <nav>. Не заменяет теги <ul> или <оl>, он просто их обрамляет. Не все группы ссылок на странице должны быть обернуты <nav>, этот элемент предназначен в первую очередь для разделов, которые состоят из главных навигационных блоков.



Рис. 6 Синтаксис семантического элемента <nav>

В качестве элементов панели навигации можно использовать не только элементы списков (Рис. 7):



Рис. 7 Пример использования элемента <nav>

* Элемент <footer>

Представляет собой нижний колонтитул содержащей его секции или корневого элемента. Обычно содержит информацию об авторе статьи, данные о копирайте и т.д. Если используется как колонтитул всей страницы, содержимое дополняется сведениями об авторских правах, ссылками на условия использования, контактную информацию, ссылками на связанное содержимое и т.п.

В одном веб-документе может быть несколько элементов <footer>. Как каждая страница, так и каждая статья может иметь свой элемент <footer>, более того, <footer> можно поместить в элемент <blockquote>, чтобы указать источник цитирования.



Рис. 8 Синтаксис семантического элемента <footer>

* Элемент <article>

Используется для группировки записей — публикаций, статей, записей блога, комментариев. Представляет собой независимый обособленный блок, предназначенный для многократного использования, как правило, начинается с заголовка. Может дублироваться на других страницах сайта и содержать внутри другие элементы <article>, которые по содержанию имеют близкое отношение к содержанию внешней статьи. Если на странице присутствует только одна статья с заголовком и текстовым содержимым, она не нуждается в обёртке элементом<article>.

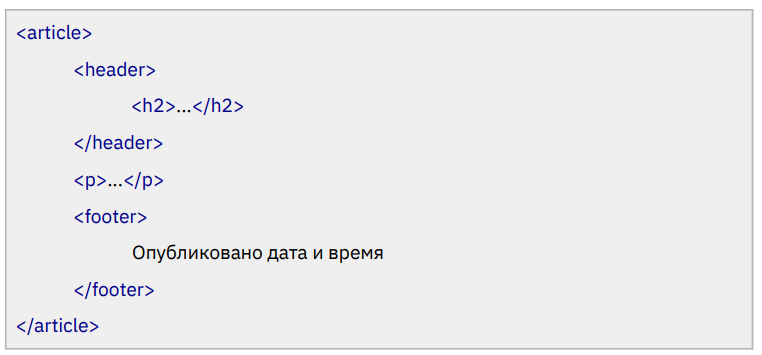


Рис. 9 Синтаксис семантического элемента <article>

* Элемент <section>

Элемент представляет собой универсальный раздел документа. Группирует тематическое содержимое, не используется многократно и обычно содержит заголовок. Не является блоком-обёрткой, для этих целей уместнее использовать элемент <div>. В качестве содержимого может выступать оглавление, разделы научных публикаций, программа мероприятия. Домашняя страница сайта также может быть поделена на секции — блок вводной информации, новости и контакты.



Рис. 10 Синтаксис семантического элемента <section>

* Элемент <aside>

Группирует содержимое, связанное с окружающим его контентом напрямую, но которое можно счесть отдельным (т.е., удаление этого блока не повлияет на понимание основного содержимого). Чаще всего элемент позиционируется как боковая колонка (как в книгах) и включает в себя группу элементов: <nav>, цифровые данные, цитаты, рекламные блоки, архивные записи. Не подходит для блоков, просто позиционированных в стороне.

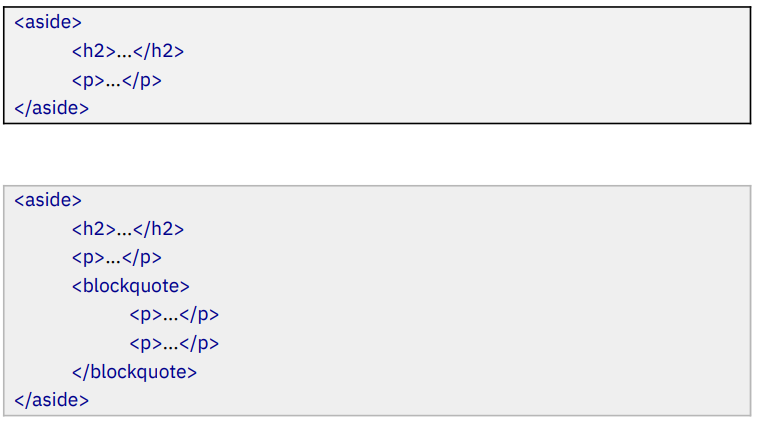


Рис. 11 Синтаксис семантического элемента <aside>

На рисунке 12 показан способ применения элемента <aside>:

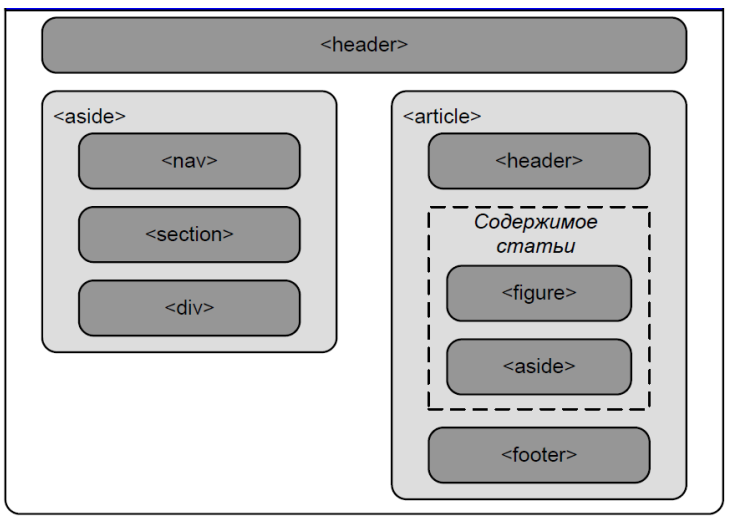


Рис. 12 Возможная структура элемента <aside>