1. Определения понятий: - Страница - это отдельный документ или файл на веб-сайте, который содержит информацию для отображения пользователю в браузере.

- Сайт - совокупность веб-страниц, связанных общей тематикой или направлением

- Веб-узел - это адрес или местоположение в сети, где располагается веб-страница.

- Веб-проект - процесс создания, разработки и поддержки веб-сайта или веб-приложения.

- Портал - веб-сайт, предоставляющий широкий спектр услуг и информации по конкретной тематике.

- Веб-ресурс - любой элемент, который может быть доступен через сеть интернет.

Этапы разработки веб-сайта:

1. Планирование и анализ: определение целей сайта, целевой аудитории, сбор требований.

2. Дизайн и макет: создание дизайна и структуры сайта, выбор цветовой палитры, шрифтов и элементов.

3. Вёрстка и разработка: кодирование, создание html, css, скриптов и других элементов сайта.

4. Тестирование и отладка: проверка работы сайта, исправление ошибок, кросс-браузерное тестирование.

5. Публикация и аналитика: загрузка сайта на хостинг, настройка аналитики для отслеживания посещаемости и поведения пользователей.

Инструменты для разработки сайта:Текстовые редакторы (например, Sublime Text, Visual Studio Code). Графические редакторы (Photoshop, Sketch). Браузеры (Chrome DevTools, Firefox Developer Tools). Фреймворки и библиотеки (Bootstrap, jQuery). CMS (Content Management System) (WordPress, Joomla). Веб-серверы (Apache, Nginx). Хостинг и FTP-клиенты (FileZilla).

2. HTML (HyperText Markup Language) - это язык разметки используемый для создания веб-страниц. Его основное назначение - структурировать содержимое веб-страницы, определять отображение различных элементов на странице. HTML использует теги для определения различных элементов страницы.

Структура HTML-документа:

- <!DOCTYPE html>: Объявление типа документа.

- <html>: Открывающий и закрывающий тег для всего документа.

- <head>: Секция с метаданными документа и внешними ресурсами.

- <title>: Заголовок страницы, отображается на вкладке браузера.

- <body>: Содержимое страницы, отображаемое в браузере.

Пример шаблона HTML-страницы:

<!DOCTYPE html>

<html>

<head>

<title>Заголовок страницы</title>

</head>

<body>

<h1>Привет, мир!</h1>

<p>Это абзац текста.</p>

</body>

</html>

3. Раздел HEAD HTML-страницы - содержит метаданные документа, такие как заголовок страницы, ссылки на внешние таблицы стилей (CSS), скрипты, метатеги и другую информацию, неотображаемую на самой странице.

Примеры содержимого раздела HEAD: Заголовок страницы: <title>Название страницы</title> Подключение внешнего CSS: <link rel="stylesheet" href="styles.css"> Подключение скриптов: <script src="script.js"></script> Метатег для указания кодировки: <meta charset="UTF-8">

4. Мета-теги языка HTML используются для предоставления информации о документе, такой как описание, ключевые слова, автор и другие метаданные. Они помогают поисковым системам и браузерам правильно интерпретировать содержимое страницы.

Примеры использования мета-тегов: Описание страницы: <meta name="description" content="Описание страницы"> Ключевые слова: <meta name="keywords" content="ключевые слова, для, поиска"> Автор: <meta name="author" content="Имя автора">

5. Теги форматирования текста языка HTML используются для изменения внешнего представления текста на веб-странице, таких как заголовки, абзацы, жирный текст, курсив и другие стилевые свойства.

Примеры тегов форматирования текста: Заголовок: <h1>Заголовок первого уровня</h1> Абзац: <p>Текст абзаца.</p> Жирный текст: <b>Жирный текст</b> Курсив: <i>Курсивный текст</i>

6. Теги работы со списками языка HTML используются для создания упорядоченных (нумерованных), неупорядоченных и списков определений.

Примеры использования тегов списков:

- Нумерованный список:

<ol>

<li>Элемент 1</li>

<li>Элемент 2</li>

</ol>

- Неупорядоченный список:

<ul>

<li>Пункт 1</li>

<li>Пункт 2</li>

</ul>

- Список определений:

<dl>

<dt>Термин</dt>

<dd>Определение</dd>

</dl>

7. Тег <div> является блочным элементом и используется для группировки других элементов страницы и создания блочной модели документа. <div> обычно используется для стилизации с помощью CSS, позиционирования элементов и облегчения работы с макетом страницы.

Пример оформления и позиционирования содержимого с помощью свойств CSS:

<div style="width: 200px; height: 200px; background-color: lightblue;"></div>

Этот пример создаст блок <div> с размерами 200x200 пикселей и голубым фоном. Стили CSS задаются через атрибут style.

8. Теги вставки изображения в языке HTML используются для добавления изображений на веб-страницу.

Примеры использования тега для вставки изображения:

<img src="путь\_к\_изображению.jpg" alt="Описание изображения">

- src - указывает путь к изображению

- alt - предоставляет альтернативный текст для случаев, когда изображение не может быть отображено

9. Гиперссылки (ссылки) в языке HTML используются для создания переходов на другие веб-страницы, внутренние якоря, адреса электронной почты или ссылки на файлы.

Примеры использования различных типов ссылок:

*<!-- Внешняя ссылка -->*

<a href="https://www.example.com">Название ссылки</a>

*<!-- Внутренняя ссылка -->*

<a href="#идентификатор\_элемента">Ссылка на элемент страницы</a>

*<!-- Ссылка на адрес электронной почты -->*

<a href="mailto:example@example.com">Написать письмо</a>

*<!-- Ссылка на архив -->*

<a href="path\_to\_archive.zip">Ссылка на загрузку архива</a>

10. Теги вставки таблиц в языке HTML используются для создания таблиц содержащих информацию в виде строк и столбцов.

Пример использования тегов таблиц:

<table>

<tr>

<th>Заголовок колонки 1</th>

<th>Заголовок колонки 2</th>

</tr>

<tr>

<td>Данные 1,1</td>

<td>Данные 1,2</td>

</tr>

</table>  
- <table> - определяет таблицу  
- <tr> - определяет строку таблицы  
- <th> - определяет ячейку заголовка  
- <td> - определяет ячейку данных

11. Теги вставки звука и видео в языке HTML используются для добавления звуковых и видеофайлов на веб-страницу.

Примеры использования тегов для вставки звука и видео:

<audio controls>

<source src="audio.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

<video controls>

<source src="video.mp4" type="video/mp4">

Your browser does not support the video element.

</video>

12. Теги вставки пользовательских форм в языке HTML используются для создания и размещения на странице форм для ввода данных пользователем, таких как текстовые поля, кнопки, флажки и др.

Примеры использования тегов для создания форм:

<form action="/submit\_form.php" method="post">

<label for="username">Имя пользователя:</label>

<input type="text" id="username" name="username">

<label for="email">Email:</label>

<input type="email" id="email" name="email">

<input type="submit" value="Отправить">

</form>  
- <form> - определяет форму  
- <input> - определяет поле ввода  
- <label> - определяет подпись для поля ввода

13. Валидация - это процесс проверки HTML и CSS кода на соответствие стандартам языка. Валидация помогает выявить ошибки и недочеты, которые могут привести к некорректному отображению страницы в браузерах.

Веб-страницы могут быть проверены на валидность с помощью онлайн-инструментов или валидаторов, предоставляемых W3C (World Wide Web Consortium). Порядок валидации документов HTML и CSS заключается в копировании кода документа в специальное поле на сайте валидатора и запуске процесса валидации. После завершения проверки валидатор предоставит отчет о найденных ошибках, которые требуется исправить.

14. CSS (Cascading Style Sheets) - это язык стилей, который используется для оформления веб-страниц, задания внешнего вида элементов HTML. Преимущества использования CSS включают легкость поддержки, удобство внесения изменений, возможность создания унифицированного дизайна на сайте.

Правила задания таблицы стилей:

- Определяются стили для элементов HTML

- Могут быть встроенными (написанными непосредственно в HTML документе), подключенными в отдельном CSS файле или заданными внутри тегов <style></style>

Примеры таблиц стилей:

*/\* Внутренние стили \*/*

<style>

body {

background-color: #f0f0f0;

font-family: Arial, sans-serif;

}

</style>

*/\* Внешний CSS файл \*/*

<link rel="stylesheet" type="text/css" href="styles.css">

*/\* Пример внешнего CSS файла \*/*

body {

background-color: #f0f0f0;

font-family: Arial, sans-serif;

}

Каскадность таблиц стилей проявляется в том, что если один и тот же элемент определен в разных таблицах стилей, браузер будет использовать правила из наиболее конкретного и специфичного источника.

15. Способы внедрения таблиц стилей в HTML-документ:

1. Встроенные стили (Inline Styles)

<p style="color: red;">Текст</p>

2. Внутренние стили (Internal Styles)

<head>

<style>

p {

color: blue;

}

</style>

</head>

3. Внешние стили (External Styles)

<link rel="stylesheet" type="text/css" href="styles.css">

16. Селектор в CSS - это часть правила стиля, которая указывает на элемент(ы) страницы, к которым будут применены стили.

Примеры различных видов селекторов в CSS:

- Стиль применяется к тегу HTML

h1 {

color: red;

}

- Универсальный класс стилей

\* {

margin: 0;

padding: 0;

}

- Псевдо-класс

a:hover {

color: blue;

}

- Псевдо-элемент

p::first-line {

font-weight: bold;

}

- Класс-идентификатор

.className#id {

font-size: 16px;

}

Пример использования разработанных таблиц стилей:

body {

font-family: Arial, sans-serif;

background-color: #f0f0f0;

}

h1 {

color: blue;

}

a:hover {

text-decoration: underline;

}  
17. Примеры использования свойств CSS:

- Фон

body {

background-color: #f0f0f0;

background-image: url('background.jpg');

}

- Цвет

p {

color: blue;

}

- Отступы

div {

margin: 10px;

padding: 5px;

}

- Поля и границы

input {

padding: 5px;

border: 1px solid #ccc;

}

18. Примеры использования свойств CSS:

- Шрифт

h1 {

font-family: Arial, sans-serif;

font-size: 24px;

font-weight: bold;

}

- Текст

p {

text-align: center;

line-height: 1.5;

}

- Списки

ul {

list-style-type: square;

}

ol {

list-style-type: decimal;

}

19. Примеры использования свойств CSS для форматирования элементов страницы:

- Таблицы

table {

border-collapse: collapse;

width: 100%;

}

td, th {

border: 1px solid black;

padding: 5px;

}

- Изображения

img {

max-width: 100%;

height: auto;

border: 2px solid #ccc;

border-radius: 10px;

}

20. Примеры использования свойств CSS на примере блока <div>:

- Границы

div {

border: 1px solid black;

border-radius: 5px;

}

- Позиционирование

div {

position: relative;

top: 50px;

left: 20px;

}

- Отступы

div {

margin: 10px;

padding: 20px;

}

21. Характеристика единиц измерения в CSS:

- Абсолютные единицы измерения (пиксели, дюймы) фиксированы и не зависят от контекста.

- Относительные единицы измерения (проценты, em, rem) зависят от других факторов, таких как размер шрифта или размер родительского элемента.

Примеры использования единиц измерения: Строка: "Arial"; Число: 12px; Проценты: 50%; Размер: 2em; Цвет: #ff0000; Адрес: url('image.jpg'); Ключевое слово: inherit

22. Основные элементы компоновки страниц сайта: Шапка (Header); Навигация(Navigation); Содержимое (Content); Боковая панель (Sidebar); Подвал (Footer)

Модульная сетка - это система разметки, в которой весь макет страницы разбивается на ряды и колонки для удобного размещения содержимого. Примеры модульных сеток могут использовать системы сеток Bootstrap, Grid CSS

23. Назначение элементов шаблона сайта:

- Шапка - верхняя часть сайта с обычно содержит логотип, название сайта, контактную информацию.

- Логотип - графическое изображение, которое представляет бренд или компанию.

- Подвал - нижняя часть сайта с обычно содержит дополнительную навигацию, ссылки на социальные сети, контактную информацию.

- Информация об авторских правах - текстовое или графическое уведомление о правах на содержимое сайта.

- Навигация - система ссылок и меню, позволяющая пользователям перемещаться по сайту.

- Основное содержимое - центральная часть сайта, где располагается основной контент.

- Контент - информация, представленная на странице сайта.

- Меню - набор ссылок для навигации по разделам сайта.

- Форма поиска - поле для ввода текста, позволяющее пользователю искать информацию на сайте.

- Форма авторизации - поле для ввода логина и пароля для доступа к персональной информации.

- Форма опроса, голосования - средство для сбора мнения пользователей.

- Карта сайта - список ссылок на все страницы сайта для удобной навигации.

Рекомендации по разработке элементов сайта:

- Создавайте шаблонный дизайн для обеспечения единообразия на всем сайте.

- Поддерживайте легкость взаимодействия с элементами, соблюдая принципы юзабилити.

- Держите информацию о компании и авторские права в подвале сайта для лучшей видимости.

24. Верстка веб-страниц - это процесс создания внешнего вида сайта с помощью HTML, CSS и иногда JavaScript. Основные способы верстки web-страниц:

- Таблицы - использование HTML-таблиц для размещения контента, считается устаревшим и не эффективным методом.

- Сеточная верстка - размещение элементов на странице с помощью сетки, созданной с помощью CSS.

- Флексбоксы - использование CSS-свойств для создания гибких и адаптивных макетов.

25. Верстка страниц сайта с помощью блоков (дивная верстка) имеет следующие достоинства:

- Гибкость и возможность легкой адаптации макета под разные разрешения экранов.

- Улучшенная доступность для поисковых систем, так как контент структурирован.

Недостатки верстки с помощью блоков:

- Возможно усложнение структуры кода из-за большого количества контейнеров.

- Необходимость аккуратного применения CSS для выравнивания и стилизации блоков.

26. Фиксированная верстка:

- Достоинства:

- Позволяет точно контролировать расположение элементов на странице.

- Легко поддерживать единообразие дизайна.

- Подходит для сайтов с фиксированной шириной контейнера.

- Недостатки:

- Не адаптивна и не подходит для различных разрешений экранов.

- Может вызывать проблемы на мобильных устройствах.

27. Резиновая верстка:

- Достоинства:

- Адаптивна и легко подстраивается под различные размеры экранов.

- Улучшает удобство пользования сайтом для пользователей.

- Позволяет создавать более гибкие макеты.

- Недостатки:

- Может вызывать проблемы с отображением на очень больших или очень маленьких экранах.

- Сложнее контролировать внешний вид элементов на странице.

28. Комбинированная верстка:

- Достоинства:

- Позволяет комбинировать фиксированные и резиновые элементы для оптимального отображения на различных устройствах.

- Обеспечивает баланс между контролем над дизайном и адаптивностью.

- Недостатки:

- Требует более сложного подхода к верстке и макетированию.

- Может потребовать дополнительного времени для разработки.

29. Адаптивная верстка:

- Достоинства:

- Макет автоматически подстраивается под различные разрешения экранов.

- Улучшает пользовательский опыт на мобильных устройствах.

- Повышает доступность сайта для широкого круга пользователей.

- Недостатки:

- Может потребовать дополнительного времени для разработки

- Не всегда легко достичь идеального отображения на всех устройствах из-за различий в размерах и пропорциях экранов.

30. Семантическая верстка - это подход к разработке сайтов, где HTML-теги используются согласно их семантическому назначению для правильной структурированности содержимого.

Структура семантического шаблона:

- <header> - верхняя часть страницы, обычно содержит логотип и навигацию.

- <nav> - навигационное меню.

- <main> - основное содержимое страницы.

- <footer> - нижняя часть страницы с дополнительной информацией.

- <aside> - содержимое, дополняющее главное содержание.

- <article> - отдельное содержимое, например, статья или пост.

- <section> - определенный раздел страницы.

Назначение основных элементов:

- Помогает поисковым системам понять структуру и смысл содержимого.

- Улучшает доступность для пользователей, использующих вспомогательные технологии.

- Повышает SEO-оптимизацию сайта за счет правильного использования тегов.

31. Правило каскадности в CSS (Cascading Style Sheets) означает, что стиль, определенный на более конкретном уровне, переопределяет стиль, определенный на более общем уровне. Это позволяет контролировать приоритетность стилей и управлять отображением элементов на веб-странице. Другие важные моменты:

- Стили, определенные непосредственно внутри тега или через атрибут style, имеют высший приоритет.

- Внешние таблицы стилей (CSS-файлы) применяются после встроенных стилей и имеют более высокий приоритет.

- Правило каскадности обеспечивает структурированность и управляемость стилей в CSS.