

ЛАБОРАТОРНАЯ РАБОТА №1

«Исследование возможностей языка разметки гипертекстов

HTML и каскадных таблиц стилей CSS»

1. Что такое HTML?

HTML (HyperText Markup Language) - это язык разметки гипертекста, который используется для создания веб-страниц и структурирования содержимого веб-сайтов.

HTML используется для определения содержимого веб-страницы, такого как текст, изображения, видео, ссылки и другие элементы, а также для создания структуры документа, определения заголовков, абзацев, списков, таблиц и других элементов.

HTML-документ состоит из текста и специальных тегов, которые определяют, какой элемент на странице должен быть отображен и как он должен быть отформатирован. Теги состоят из открывающего и закрывающего элементов, и между ними располагается содержимое элемента.

HTML является одним из основных языков, используемых в веб-разработке, и является ключевым инструментом для создания веб-страниц. HTML используется совместно с другими технологиями, такими как CSS и JavaScript, для создания более сложных и интерактивных веб-сайтов.

2. Для чего в HTML используются теги?

Теги в HTML используются для определения и оформления элементов веб-страницы. Каждый тег обозначает определенный элемент на странице и задает его свойства и атрибуты. Например, тег `<p>` используется для определения абзаца текста, а тег `` используется для вставки изображения на страницу.

Теги в HTML имеют две основные части - открывающий и закрывающий элементы, которые обрамляют содержимое элемента. Например, открывающий тег `<p>` обозначает начало абзаца текста, а закрывающий тег `</p>` обозначает конец абзаца.

Кроме того, HTML имеет также и одиночные теги, которые не имеют закрывающего элемента и используются для вставки элементов, которые не имеют содержимого. Например, тег `
` используется для создания переноса строки.

Теги в HTML играют важную роль в создании веб-страниц и позволяют определить различные элементы и их свойства, такие как шрифт, размер, цвет, выравнивание и другие параметры. Кроме того, использование правильных тегов и атрибутов является важным для создания доступных и понятных веб-сайтов для пользователей и поисковых систем.

3. Расскажите о структуре HTML-документов?

Структура HTML-документа обычно состоит из двух основных частей: `head` (шапка документа) и `body` (основное содержимое документа).

`Head` - это часть документа, которая содержит метаданные, такие как заголовок страницы, ключевые слова, описание страницы, ссылки на стили, скрипты и другие метаданные, которые не отображаются на странице, но важны для ее правильной работы и SEO-оптимизации.

`Body` - это часть документа, которая содержит основное содержимое страницы, такое как текст, изображения, таблицы, ссылки и другие элементы.

В HTML есть несколько тегов, которые являются обязательными и должны присутствовать в каждом документе, чтобы он был корректным.

- Тег `<!DOCTYPE>` - это первый тег в документе, который определяет тип документа. Это не является HTML-тегом, но обязательно присутствует в документе.
- Тег `<html>` - определяет начало HTML-документа и его конец `</html>`. Все элементы HTML должны находиться внутри тега `<html>`.
- Тег `<head>` - определяет метаданные документа, такие как заголовок страницы, мета-теги, скрипты и стили. Тег `<head>` обязателен, даже если в нем нет содержимого.
- Тег `<title>` - определяет заголовок страницы, который отображается во вкладке браузера и является важным элементом SEO.

- Тег `<body>` - определяет основное содержимое страницы, такое как текст, изображения, таблицы, ссылки и другие элементы.

4. Для чего в HTML-документах используется элемент META?

Элемент META в HTML-документах используется для указания метаданных, то есть информации, которая не отображается на веб-странице, но содержит дополнительные сведения о документе. Эти метаданные используются браузерами, поисковыми системами и другими инструментами для более эффективной работы с веб-страницами.

Наиболее распространенными атрибутами элемента META являются:

- "charset" - указывает кодировку документа
- "viewport" - определяет параметры масштабирования и отображения страницы на мобильных устройствах
- "description" - описание содержимого страницы для поисковых систем
- "keywords" - ключевые слова, связанные с содержанием страницы
- "author" - имя автора документа
- "robots" - инструкции для поисковых роботов, указывающие, следует ли индексировать страницу и какие ссылки следует следовать

В целом, элемент META является важным инструментом для оптимизации веб-страницы для поисковых систем, а также для обеспечения корректной работы с ней в различных браузерах и на различных устройствах.

5. Как в HTML-документах реализуются заголовки частей текста?

В HTML для обозначения заголовков используются теги от `h1` до `h6`. Номер тега указывает уровень заголовка, при этом `h1` обозначает наиболее важный заголовок, а `h6` - наименее важный.

6. Как вставить в HTML-документ преформатированный текст?

Для вставки преформатированного текста в HTML-документе используется тег `<pre>`. Этот тег позволяет отображать текст с сохранением всех пробелов, переносов строк и других пробельных символов.

7. В чем преимущества использования каскадных таблиц стилей в HTML?

Использование каскадных таблиц стилей (CSS) в HTML имеет множество преимуществ, вот некоторые из них:

Раздельное хранение стилей и содержимого: CSS позволяет разделять стили и содержимое веб-страницы. Это упрощает поддержку и изменение внешнего вида сайта, так как вам не нужно редактировать каждую страницу отдельно.

Большая гибкость и контроль: CSS предоставляет больше гибкости и контроля над оформлением веб-страницы. Вы можете изменять стили для отдельных элементов, создавать классы стилей и применять их к нескольким элементам, задавать стили для различных состояний элементов (например, при наведении курсора на ссылку).

8. Где возможно размещение информации о стилях в HTML-документе?

Информация о стилях в HTML-документе может быть размещена в нескольких местах:

Внутри тега `<head>` с помощью тега `<style>`. Этот способ позволяет определить стили, которые будут применяться только к данной странице.

Html.

Внутри отдельного файла с расширением `.css`, который затем подключается к HTML-документу с помощью тега `<link>`. Этот способ позволяет использовать одни и те же стили на нескольких страницах сайта.

```
<head>
  <title>Заголовок страницы</title>
  <link rel="stylesheet" href="style.css">
</head>
```

9. Что такое URL? Расскажите о формате URL?

URL (Uniform Resource Locator) - это адрес, по которому можно найти ресурс в Интернете. Это может быть веб-страница, изображение, видео и т.д.

Формат URL состоит из следующих частей:

<протокол>://<имя_хоста>:<порт>/<путь>?<параметры>#<якорь>

- Протокол - это соглашение о том, каким образом будут передаваться данные. Например, протокол HTTP используется для передачи веб-страниц, протокол FTP для передачи файлов, протокол SMTP для передачи почты.
- Имя хоста - это уникальное имя, которое идентифицирует компьютер, на котором расположен ресурс. Например, для веб-сайта "www.example.com" имя хоста будет "www".
- Порт - это номер, который идентифицирует конкретный процесс на сервере. По умолчанию используется порт 80 для протокола HTTP и порт 443 для протокола HTTPS.
- Путь - это конкретный адрес ресурса на сервере. Например, для страницы "about.html" путь может быть "/about.html".
- Параметры - это дополнительная информация, которая передается на сервер вместе с запросом. Они передаются после знака вопроса и разделяются знаком амперсанда (&). Например, параметры могут использоваться для передачи поискового запроса.
- Якорь - это ссылка на конкретный элемент страницы. Якорь передается после знака решетки (#). Например, якорь может использоваться для ссылки на конкретный раздел страницы.

10. Как задаются ссылки в HTML-документе?

Ссылки в HTML-документе задаются с помощью тега <a> (anchor), который обозначает гиперссылку.

Синтаксис тега <a> следующий:

Текст ссылки

Атрибут href определяет URL-адрес ресурса, на который ведет ссылка. Вместо URL необходимо указать адрес страницы, на которую должна вести ссылка.

Текст ссылки заключается между открывающим и закрывающим тегами <a>.

При клике на ссылку, браузер перенаправляет пользователя на страницу, указанную в атрибуте href. Если вместо URL-адреса в атрибуте href указать якорь (#), то ссылка будет переводить на элемент страницы, на который ссылается якорь.

```
<a href="#contacts">Контакты</a>

...

<h2 id="contacts">Контакты</h2>
```

11. Приведите пример организации списков в HTML?

Существуют три типа списков в HTML: неупорядоченный список (), упорядоченный список () и список определений (<dl>).

1. Неупорядоченный список () - список, элементы которого не имеют определенного порядка. Каждый элемент списка обозначается тегом . Пример:

html

Copy code

```
<ul>
  <li>Первый элемент</li>
  <li>Второй элемент</li>
  <li>Третий элемент</li>
</ul>
```

2. Упорядоченный список () - список, элементы которого имеют порядковый номер. Каждый элемент списка обозначается тегом . Пример:

html

Copy code

```
<ol>
  <li>Первый элемент</li>
  <li>Второй элемент</li>
  <li>Третий элемент</li>
</ol>
```

3. Список определений ('<dl>') - список, элементы которого состоят из термина (слово или фраза) и его определения. Каждый термин обозначается тегом '<dt>', а его определение - тегом '<dd>'. Пример:

html

Copy code

```
<dl>
  <dt>HTML</dt>
  <dd>Язык разметки гипертекста</dd>
  <dt>CSS</dt>
  <dd>Каскадные таблицы стилей</dd>
</dl>
```

При использовании списков можно комбинировать различные типы списков и вкладывать один список в другой. Например, упорядоченный список может содержать вложенные неупорядоченные списки и наоборот.

12. Как вставить графический элемент в HTML-документ?

Для вставки графического элемента в HTML-документ используется тег ''.

Синтаксис тега '':

html

Copy code

```

```

Атрибут '<src>' указывает URL изображения, который может быть как локальным, так и удаленным. Атрибут '<alt>' определяет текст, который будет отображаться в случае, если изображение не может быть загружено или если пользователь работает с устройством, не поддерживающим изображения.

13. Какие атрибуты используются в тегах таблиц HTML?

(ТЕГИ)

1. '<table>' - создает таблицу в HTML-документе.
2. '<tr>' - определяет строку таблицы.
3. '<td>' - определяет ячейку таблицы.

1. ``border``: задает толщину границы вокруг таблицы и ее ячеек.
2. ``cellspacing``: задает расстояние между ячейками таблицы.
3. ``cellpadding``: задает расстояние между содержимым ячеек таблицы и их границами.
4. ``width``: задает ширину таблицы.
5. ``height``: задает высоту таблицы.
6. ``align``: выравнивает таблицу по горизонтали.
7. ``valign``: выравнивает таблицу по вертикали.
8. ``bgcolor``: устанавливает цвет фона таблицы.
9. ``colspan``: определяет количество объединяемых ячеек в строке.
10. ``rowspan``: определяет количество объединяемых ячеек в столбце.
11. ``scope``: указывает, на какую область данных таблицы относится ячейка.
12. ``headers``: связывает ячейку с заголовком таблицы.
13. ``id``: идентифицирует элемент таблицы для целей стилизации или скриптов.
14. ``class``: определяет класс элемента таблицы для целей стилизации.
15. ``style``: определяет стили элемента таблицы.
16. ``title``: добавляет всплывающую подсказку к элементу таблицы.
17. ``lang``: определяет язык элемента таблицы.
18. ``dir``: определяет направление текста элемента таблицы.
19. ``accesskey``: определяет клавишу доступа для элемента таблицы.
20. ``tabindex``: определяет порядок перехода между элементами таблицы при нажатии клавиши "Tab".

14. В чем состоят основные отличия HTML и XHTML?

Синтаксис: HTML имеет более свободный синтаксис, в то время как XHTML более строго следует правилам XML, включая синтаксические требования, такие как закрытые теги и правильная вложенность.

Документы XHTML должны быть "валидными" в соответствии с определенной DTD (Document Type Definition), что означает, что документ должен соответствовать определенному стандарту и не содержать ошибок. В то время как в HTML допускаются ошибки и несоответствия стандартам.

В XHTML необходимо строго определять пространство имен и использовать корректный синтаксис для всех элементов, что обеспечивает более строгую структуру и читаемость документа.

XHTML требует более строгого разделения содержимого и представления, чем HTML. Это означает, что XHTML документы должны содержать только описательную структуру документа, в то время как все аспекты представления должны быть определены в отдельных таблицах стилей.

В XHTML отсутствует ряд элементов, которые были доступны в HTML, например, элементы фреймов (<frame>, <frameset>) и несколько атрибутов.

XHTML поддерживает XML-совместимые технологии, такие как XSLT (eXtensible Stylesheet Language Transformations) и XPath (XML Path Language), которые обеспечивают более мощные возможности для обработки и преобразования содержимого документов.

Несмотря на то, что XHTML имеет более строгий и более сложный синтаксис, он также обладает рядом преимуществ, таких как лучшая поддержка современных стандартов веб-разработки, более эффективный и безопасный способ создания веб-страниц.

15. Что понимают под HTML вёрсткой?

HTML-верстка - это процесс создания веб-страницы с использованием языка гипертекстовой разметки HTML. HTML используется для определения структуры страницы и размещения на ней различных элементов, таких как текст, изображения, видео, формы и другие элементы.

HTML-верстка включает в себя создание разметки страницы, определение стилей, управление расположением и взаимодействием элементов на странице, а также определение многих других аспектов внешнего вида и функциональности страницы.

HTML-верстка может выполняться вручную, используя текстовый редактор, или с помощью специализированных инструментов, таких как программы-редакторы, фреймворки и библиотеки для веб-разработки. В процессе HTML-верстки

учитываются различные факторы, такие как соответствие стандартам HTML и CSS, совместимость с различными браузерами, доступность и удобство использования для конечного пользователя.

16. Что такое валидация HTML?

Валидация HTML - это процесс проверки соответствия веб-страницы стандартам языка гипертекстовой разметки HTML. В процессе валидации происходит проверка наличия и правильности использования тегов, атрибутов, значений и других элементов, которые используются в HTML-разметке.

, что веб-страница корректно отображается в различных браузерах, и обеспечить соответствие стандартам HTML, определенным организацией W3C (World Wide Web Consortium).

Валидация HTML важна, поскольку помогает обнаружить и исправить ошибки, которые могут привести к некорректному отображению страницы, проблемам с доступностью и индексации страницы поисковыми системами. Кроме того, валидная HTML-разметка может помочь улучшить производительность и оптимизацию сайта.

Для валидации HTML существуют специальные инструменты, например, W3C Markup Validation Service, которые позволяют проверить веб-страницу на соответствие стандартам HTML и получить отчет о найденных ошибках и предупреждениях.

17. Для чего используются формы в языке HTML?

Формы (form) в языке HTML используются для создания интерактивных элементов на веб-странице, которые позволяют пользователю вводить данные и отправлять их на сервер для обработки. Формы могут содержать такие элементы как текстовые поля, кнопки, переключатели, флажки, списки и другие элементы.

Основное назначение форм - это получение информации от пользователей, которую можно использовать для различных целей, например:

- регистрации пользователей;

- отправки сообщений и запросов на сервер;
- поиска информации на сайте;
- оформления заказов и покупок;
- заполнения анкет и опросов;
- других целей.

Для того, чтобы формы работали на веб-странице, нужно указать соответствующие атрибуты и настроить обработку данных на сервере. При отправке формы на сервер происходит передача данных в виде параметров HTTP-запроса, которые можно обработать с помощью серверного языка программирования, например, PHP, Python, Ruby и других.

18. Какие атрибуты имеет тег <FORM> и для чего они используются?

Тег <form> имеет несколько атрибутов, которые используются для настройки параметров формы. Некоторые из наиболее распространенных атрибутов:

- action - определяет адрес скрипта или страницы, которая будет вызвана при отправке формы.
- method - определяет метод отправки данных формы на сервер (обычно GET или POST).
- name - задает имя формы для обращения к ней из скрипта на стороне клиента или сервера.
- id - задает уникальный идентификатор формы для использования в CSS и JavaScript.
- enctype - определяет тип кодирования данных формы (обычно "application/x-www-form-urlencoded" или "multipart/form-data" для отправки файлов).
- target - определяет, где будет открыт результат отправки формы (обычно _self для открытия в текущем окне или _blank для открытия в новом окне).

19. В чем отличие методов отправки формы GET и POST?

Метод GET и метод POST являются двумя основными методами отправки данных формы на сервер.

GET-запрос используется для передачи данных на сервер через URL-адрес. Данные, отправляемые с помощью метода GET, добавляются в конец URL-адреса в виде параметров запроса. GET-запрос используется в основном для получения данных с сервера и не должен использоваться для отправки конфиденциальной информации, так как параметры запроса могут быть видны в адресной строке браузера.


POST-запрос отправляет данные формы в теле запроса HTTP-запроса на сервер. Параметры запроса не отображаются в URL-адресе, поэтому метод POST обычно используется для отправки конфиденциальной информации, такой как пароли и логины. POST-запрос также может быть использован для отправки больших объемов данных.

Основное отличие между методами GET и POST заключается в том, что метод GET имеет ограничение на длину URL-адреса и может быть кэширован браузером, в то время как метод POST не имеет ограничений на длину запроса и не кэшируется браузером. Кроме того, POST-запрос может использоваться для отправки файлов на сервер, что не возможно с помощью метода GET.

20. Как реализовать элемент ввода многострочного текста в форме?

Для реализации элемента ввода многострочного текста в форме используется тег `<textarea>`. Он позволяет пользователю вводить текст в несколько строк и имеет следующий синтаксис:

html

 Copy code

```
<textarea rows="4" cols="50">
```

Введите текст здесь...

```
</textarea>
```

Атрибут `rows` задает количество строк, атрибут `cols` — количество столбцов. Между открывающим и закрывающим тегами `<textarea>` можно указать начальный текст, который будет отображаться внутри поля ввода до тех пор, пока пользователь не начнет вводить свой текст.

21. Как реализовать элемент ввода типа переключатель в форме?

Для реализации элемента ввода типа переключатель (радиокнопки) в форме используется тег `<input>` с атрибутом `type="radio"`. Каждая радиокнопка должна иметь уникальное значение атрибута `name`, чтобы группировать их вместе, и значение атрибута `value`, которое будет отправляться на сервер в случае выбора этой кнопки. Пример кода для создания группы радиокнопок:

```
<form>
  <input type="radio" id="radio1" name="gender" value="male">
  <label for="radio1">Мужской</label><br>
  <input type="radio" id="radio2" name="gender" value="female">
  <label for="radio2">Женский</label><br>
  <input type="radio" id="radio3" name="gender" value="other">
  <label for="radio3">Другое</label>
</form>
```

22. Для чего используется тег `<SELECT>`?

Тег `<select>` в HTML используется для создания элемента формы, который позволяет пользователю выбрать один или несколько вариантов из выпадающего списка. Он создает список опций, которые могут быть выбраны пользователем, и предоставляет удобный интерфейс для выбора одного или нескольких вариантов.

Тег `<select>` содержит один или несколько элементов `<option>`, которые определяют каждый вариант, который пользователь может выбрать. Также в теге `<select>` могут быть указаны атрибуты, которые определяют свойства списка, такие как множественный выбор, размеры и т.д.

```
<form>
  <label for="colors">Выберите цвет:</label>
  <select id="colors" name="colors">
    <option value="red">Красный</option>
    <option value="green">Зеленый</option>
    <option value="blue">Синий</option>
  </select>
</form>
```

23. Как выполняется группировка элементов списка выбора в выпадающих списках?

Для группировки элементов списка выбора в выпадающих списках в HTML используется тег `<optgroup>`. Этот тег позволяет создавать группы элементов списка внутри тега `<select>`. Например, вот как может выглядеть код с использованием тега `<optgroup>`:

```
<select>
  <optgroup label="Цвета">
    <option>Красный</option>
    <option>Зеленый</option>
    <option>Синий</option>
  </optgroup>
  <optgroup label="Фрукты">
    <option>Яблоко</option>
    <option>Груша</option>
    <option>Банан</option>
  </optgroup>
</select>
```

24. Приведите пример HTML-формы для ввода информации о студенте: ФИО, группа, № зачетной книжки?

```
<form action="submit-form.php" method="POST">
  <label for="fio">ФИО:</label>
  <input type="text" id="fio" name="fio" required>

  <label for="group">Группа:</label>
  <input type="text" id="group" name="group" required>

  <label for="student-id">№ зачетной книжки:</label>
  <input type="text" id="student-id" name="student-id" required>

  <button type="submit">Отправить</button>
</form>
```