**Создание сайта средствами HTML-редактирования и размещение на нем мультимедиа контента.**

**25.01.2022г.**

**Тема 1.1 Создание веб- страницы (1 ч.)**

Web- страницы выглядят зачастую очень пестро: разнокалиберные куски текста, таблицы, картинки, врезки, сноски и даже фильмы. Но описывается все это в виде обычного текста. Для форматирования содержимого web- страниц применяется особый язык HTML. С помощью команд - тегов - этого языка создаются абзацы текста, и заголовки, и врезки, и даже таблицы. Теги помещаются между символами < и >. Так же между символами < и >, помимо имени тега, присутствуют еще какие - то данные, которые называются атрибутами тега, задающие его параметры. Каждый атрибут тега имеет имя, за которым ставится знак равенства, и значение данного атрибута, взятое в двойные кавычки. Атрибуты тегов бывают обязательными и необязательными. Обязательные атрибуты должны присутствовать в теге в обязательном порядке. Необязательные атрибуты могут быть опущены; в таком случае тег ведет себя так, будто соответствующему атрибуту присвоено значение по умолчанию

Основные правила, согласно которым пишется HTML- код:

* Имена тегов и имена атрибутов можно писать как прописными (большими), так и строчными (маленькими) буквами. Традиционно в языке HTML имена атрибутов тегов пишутся прописными буквами, а их значения - строчными, если, конечно, значение не чувствительно к регистру букв.
* Между символами <, >, / и именами тегов, а также внутри имен тегов не допускаются пробелы и переносы строки.
* В обычном тексте, не являющемся тегом, не должны присутствовать символы < и >. (эти символы называют недопустимыми). В противном случае Web – обозреватель сочтет фрагмент текста, где встречается один из этих символов, тегом и отобразит web - страницу некорректно.
* Имена атрибутов тегов пишутся между символами < и > после имени тега и отделяют от него пробелом или разрывом строки. Если в теге присутствуют несколько атрибутов, их отделяют друг от друга также пробелами или разрывами строки.
* Внутри имен атрибутов не должны присутствовать пробелы, в противном случае Web - обозреватель посчитает, что это не один атрибут, а несколько.
* Значение атрибута тега пишут после его имени и заключают в двойные кавычки. Между именем атрибута тега и его значением ставят знак равенства.
* Между именем атрибута тега, знаком равенства и открывающими кавычками могут присутствовать пробелы или разрывы строк.
* Символы двойных кавычек недопустимы и не должны присутствовать в обычном тексте, иначе web- обозреватель посчитает следующий за ним текст значением атрибута тега.

Когда web- браузер получает документ, он определяет, как документ должен быть интерпретирован. Самый первый тег, который встречается в документе, должен быть тегом <html>. Данный тег сообщает web- браузеру, что ваш документ написан с использованием HTML. МинимальныйHTML- документтак:

<html>

<head>

<title>…… </title>

</head>

<body>

.

.

.

</body>

</html>

Тег заголовочной части документа должен быть использован сразу после тега <HTML> и более нигде в теле документа. Данный тег представляет общее описание документа.

Большинство web- браузеров отображает содержимое тега <title>в заголовке окна, содержащего документ, и в файле закладок, если он поддерживается web- браузером. Заголовок, ограниченный тегами <title> ..</title>, размещается между внутри <head> - тегов. Заголовок документа не появляется при отображении самого документа в окне.

Теги тела идентифицируют отображаемые в окне компоненты HTML- документа. Тело документа может содержать ссылки на другие документы, текст и другую форматированную информацию. Тело документа должно находится между тегами <body></body>. Эта часть документа, которая отображается как текстовая и графическая (смысловая) информация вашего документа.

Теперь представим схему первого документа в виде кода HTML:

<HTML>

<HEAD>

<TITLE>Мояперваястраница</TITLE>

</HEAD>

<BODY>

</BODY>

</HTML>.

**Элементы заголовка**  организованы по уровням от h1 до h6. Текст, содержащийся в элементе заголовка, воспринимается браузером как «блок» (блочное отображение). Кроме того, сверху снизу он отделяется пустым пространством (которое иногда называют «пробельный символ»). Размер текста наибольший для h1 (заголовок первого уровня) и наименьший для h6 (заголовок шестого уровня). Взависимости от используемого шрифта, текст, содержащейся в элементах h4,h5 и h6, может быть выглядеть меньшим, чем установленный по умолчанию. Весь текст, окруженный тегами заголовка, отображается полужирным шрифтом.

**Элемент абзаца**  применяется для группировки предложений и разделов текста. Текст, окруженный тегами <p> и </p>, отображается одним «блоком» (так называемое блочное отображение). Сверху и снизу от него остается пустое пространство.

**Элементы разрыва строки**  используется, чтобы принудительно перенести текст на новую строку перед отображением следующего элемента или фрагмента текста на веб- странице в браузере. Этот элемент разрыва строки не используется в виде пары открывающего и закрывающего тегов. Это одиночный или автономный тег. В синтаксисе элемент разрыва строки выглядит как <br>.

**Выравнивание текста.**  Вы заметили, что заголовки и текст начинаются от левой границы. Это называется выравниванием по левому краю и являются выравниваем по умолчанию для веб - страниц. Иногда необходимо выровнять абзац текста или заголовок по центру или по правому краю (по ширине). Для этого можно использовать атрибут align. Цель атрибута - в изменении свойств HTML- элементов. В данном случае атрибут align меняет горизонтальное выравнивание элемента (левое, по центру или правое) на веб – странице. Чтобы выровнять элемент по центру, используйте атрибут со значением align= “center”.

Чтобы выровнять элемент по правому краю веб - страницы, используйте align =”right”.Итак, по умолчанию выравнивание установлено по левому краю. Атрибут alignможет использоваться для множества блочных элементов, в том числе абзаца (p) и заголовка (от h1 до h6). Атрибут выравнивание не применяется в версии HTML5.

**Для определения размера, цвета и семейства шрифта** используется тег<Font></Font>.

<Fontsize= число color= цвет face=имя шрифта (семейства)>

Атрибуты size=x - размер шрифта. Целое число от 1 до 7, причем 7 соответствует самому большому шрифту.

Face= имя шрифта. Использовать рекомендуют шрифтовые семейства:

* Serif (обычноэтоTimes New Roman);
* Sans-Serif (Arial, Verdana, Tahoma, Geneva);
* Monospaced(Courier New).

В сложных случаях используют синтаксис:

**<Font face=”Balica, TimesET, Serif” >**

Color=color. Цвет текста.

Названия цветов и их значение:

|  |  |  |  |
| --- | --- | --- | --- |
| Цвет | Код | Цвет | Код |
| Aqua(морская волна) | #00ffff | Black(черный) | #000000 |
| Grey (серый) | #808080 | Green (зеленый) | #008000 |
| Navy (темно - синий) | #000080 | Olive (оливковый) | #808000 |
| Silver (серебрянный) | #c0c0c0 | Teal (бирюзовый) | #008080 |
| Blue(синий) | #0000ff | Fuchsia (фуксия) | #ff00ff |

**Элемент цитирования.**

Кроме организации текста в абзацы и заголовки, иногда вам понадобится добавить на веб- страницу цитату. **Элемент цитирования**  предназначен для отображения блока текста цитаты в особой форме - с отступом от левой и правой границ. Блок текста с отступом начинается с тега <blockquote>и заканчивается тегом </blockquote>

**Элементы логического стиля,**  которые также называют  **элементами фразы**, определяют логический стиль, используемый для отображения текста между контейнерными тегами. Браузеры по - разному интерпретируют этот стиль. Элементы логического стиля отображаются в одной строке с текстом (строчное отображение) и могут применяться к разделу текста или даже к одному символу. Например, элементstrong указывает, что текст, связанный с ним, будет отображаться в «акцентированном» виде по отношению к нормальному тексту на странице.

Рассмотрим часто встречающие элементы логического стиля и примеры их использования.

|  |  |  |
| --- | --- | --- |
| **Элемент** | **Пример** | **Использование** |
| Abbr | ФБР | Обозначает текст как аббревиатуру; конфигурирует атрибут заголовка, содержащий полное название |
| B | **Полужирный**  шрифт | Текст, не являющийся исключительно важным, но традиционно выделяемый полужирным шрифтом |
| Cite | *Цитированный* текст | Обозначает цитату или ссылку; как правило, отображается курсивом |
| Code | Текст  **кода** | Обозначает образцы программного кода; как правило, шрифт с фиксированным пробелом |
| Dfn | *Определение* | Обозначает определение или термин; как правило, отображается курсивом |
| Em | *Выделенный*  текст | Выделяет необходимый текст по отношению к остальному тексту; как правило, отображается курсивом |
| I | *Выделенный*  текст | Текст, не являющийся исключительно важным, но традиционно выделяемый курсивом |
| Kbd | **Набираемый**  текст | Обозначает текст, набираемый с клавиатуры; как правило, шрифт с фиксированным пробелом |
| Mark | Выделенный текст | Выделяемый текст, который легко заметить |
| Samp | **Образец** | Показывает пример выходных данных программы; как правило, шрифт с фиксированным пробелом |
| Small | Маленький текст | Мелкий шрифтом отображаются сведения об авторских правах и сноски |
| Strong | **Акцентированный**  текст | Выделяет необходимый текст из окружающего текста; как правило, отображается полужирным начертанием |
| Sub | Подстрочный текст | Отображает нижний индекс в виде мелкого текста ниже уровня строки |
| Sup | Надстрочный текст | Отображает верхний индекс в виде мелкого текста ниже уровня строки. |
| var | *переменная* | Обозначает и отображает переменную или выходные данные программы; как правило, отображается курсивом |

Элементы логического стиля - это контейнеры, поэтому необходимо использовать открывающий и закрывающий теги.

Используя тег<hr>, возможно поместить в текст **горизонтальную черту**.

<**hr size**= число**width**=число/% **align**= left/right/center **noshade**>

Параметры тега <hr>.

Size- толщина линии в пикселах.

Width– ширина линии в пикселах (фиксированная ширина) или процентах от ширины окна браузера.

Align- расположение на экране (слева/по центру / справа).

Noshade- по умолчанию линия представлена в 3D –виде с тенью. Флаг Noshade позволяет представить линию просто однотонной окрашенной полоской.

**Задание № 1. Создание простейшего файла HTML**

1. Создайте личную папку, куда вы будете сохранять все файлы своего сайта.

2. Запустите программу Блокнот (Notepad).

3. Наберите в окне программы простейший файл HTML. Расписание занятий на вторник

4. Сохраните файл под именем RASP.HTML (обязательно укажите тип файла HTML при сохранении) в личной папке.

5. Для просмотра Web-страницы используйте любую программу браузера (Internet Explorer, Opera, Mozilla Firefox или другую). Для этого, не покидая программу Блокнот (сверните окно на панель задач), откройте личную папку и двойным кликом по файлу RASP.HTML откройте окно браузера.

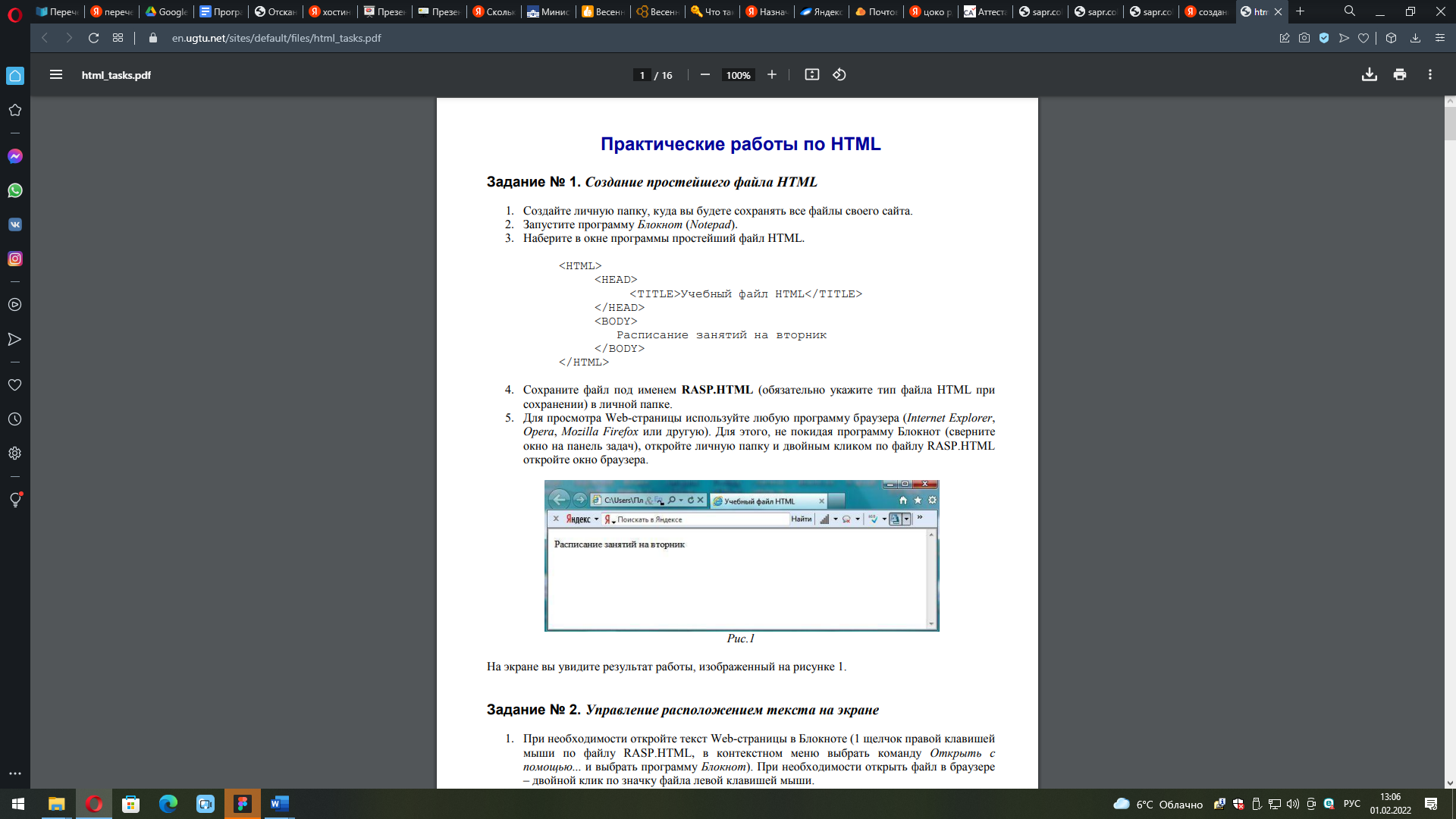


Рис.1 На экране вы увидите результат работы, изображенный на рисунке 1.

**25.01.2022г.**

**Тема 1.2 Организация внутри HTML- документа. (1ч.)**

Списки используются на веб - страницах для организации информации. Когда вы верстаете веб - страницы, помните, что заголовки, короткие абзацы и списки упрощают восприятие и чтение контента, размещенного на вашей странице. В языке HTML используются три типа списка: списки определений, упорядоченные списки и неупорядоченные списки. В  **неупорядоченных списках** отображаются маркеры перед каждым пунктом списка. Этот маркер может быть нескольких типов: диск (по умолчанию), квадрат и круг.

Неупорядоченные списки начинаются тегом <ul> и заканчивается тегом </ul>. Каждый элемент списка начинается тегом <li> и заканчивается тегом </li>.

Пример:

<h1>Популярные веб - серверы </h1>

<ul>

<li>Веб- серверApache</li>

<li>Microsoft IIS </li>

<li>Веб - серверOracle iPlanet</li>

</ul>

Для смены типа маркера используется атрибут type. Например, чтобы создать неупорядоченный список, оформленный квадратными маркерами, используя код ultype=”square”.

Circle

Disk.

**Упорядоченные списки**  используют систему нумерации или букв для организации информации, содержащейся в списке. Упорядоченный список может быть организован с помощью чисел (по умолчанию), заглавных букв, строчных букв, заглавных римских чисел и строчных римских чисел.

Упорядоченные списки начинаются с тегом <ol> и заканчиваются тегом </ol>. Каждый элемент списка начинается с тегом <li> и заканчивается тегом </li>.

Атрибут type используется для изменения символа нумерации списка. Например, чтобы создать упорядоченный список, организованный заглавными буквами, используйте код oltype=”A”

1 - числа (по умолчанию).

А – Заглавные буквы

а - строчные буквы

I- римские числа

i –строчные римские числа.

Атрибут start полезен, если вам нужно, чтобы список начинался с целого числа, но не с 1 (например, start =”10”). Чтобы маркеры списка отображались в обратном порядке, примените новый атрибут reversed(setreversed=”reversed”).

**Списки определений**  помогают организовать термины и их определения. Термины выделяются, а их определения могут быть любой длины, необходимой, чтобы передать смысл. Каждый определяемый термин начинается на отдельной строке у границы. Каждое определение начинается на отдельной строке и имеет отступ. Списки определений также удобны для организации часто задаваемых вопросов и ответов на них. Вопросы и ответы смещаются на отступ. Любой тип информации, которая состоит из списка соответствующих терминов и более длинных определений, хорошо подходит для организации с помощью списка определений.

Списки определений начинается с тега <dl> и заканчивается тегом </dl>. Каждый термин списка начинается с тега <dt> и заканчивается тегом </dt>. Каждое определение термина (определение данных) начинается с тега <dd>и заканчивается тегом </dd>.

Пример:

<dl>

<dt>TCP</dt> (термин)

<dd> Протокол управления передачей – это метод (протокол), используемый совместно с межсетевым протокол …….. (определение)

</dd>

</dl>.

Добавить на Web- страницу **графическое изображение** позволяет ординарный тег <img>. Web- обозреватель поместит изображение в том месте Web- страницы, в котором встретится тег <img>.

Обязательный атрибут тега src служит для указания интернет- адреса файла с изображением.

ПРИМЕР:

<imgsrc=”image.gif”>

Этот тег помещает на Weи - страницу изображение, хранящееся в файле image.gif, который находиться в той же папке, что и файл самой этой Web- страницы.

Поскольку изображение хранится в отдельном от Web- страницы файле, Web- обозревателю придется послать Web-серверу еще один запрос на его получение. Web- серверу нужно найти этот файл и отправить его Web- обозревателю. Файл должен загрузиться по сети.

На все это требуется время. Если изображений на Web- странице много, все они велики по размеру, а канал связи медленный, понадобиться значительное время. Может случиться так, что сама Web- страница будет успешно загружена и отображена на экране, а изображения - еще нет. И Web- обозреватель вместо не загруженного еще изображения выведет на экран пустой прямоугольник.

Возникают две проблемы. Во – первых, пустые прямоугольники вместо изображений выглядят некрасиво. Во- вторых, посетитель не сможет понять, что за изображение должно находиться вместо того или иного прямоугольника, и стоит ли ждать окончания его загрузки.

И если с первой проблемой справиться практически невозможно, то вторую вполне можно решить. Для этого тег <img> поддерживает необязательный атрибут alt, с помощью которого указывается так называемый текст замены. Он будет отображаться в пустом прямоугольнике, обозначающем незагруженное изображение, пока это изображение не загрузиться.

**Выравнивание текста по краю изображения.**  По умолчанию, когда изображение вставляется в строку текста, строка выравнивается по низу изображения. Изменить эту установку можно при помощи атрибута Align тега <img>. В таблице описаны значения атрибута Align.

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| Top | Выравнивание текста по верху изображения |
| Middle | Выравнивание текста по середине изображения |
| Botton | Выравнивает текст по низу изображения |

**Позиционирование изображения на странице**

По умолчанию программа просмотра выводит изображение в текущей строке. Текст не обтекает его. Однако при помощи атрибута Align тега <img> изображение можно сделать «плавающим», т.е. заставить текст расположиться вокруг изображения.

|  |  |
| --- | --- |
| **Значение** | **Описание** |
| Left | Обтекаемое текстом изображение прижато к левой стороне окна браузера |
| Right | Обтекаемое текстом изображение прижато к правой стороне окна браузера. |

При помощи тега <img>программе просмотра можно сообщить размеры изображения, которое затем размещается на странице. Рекомендуется использовать эту возможность из следующих соображений. Это позволяет читателю, отключившему загрузку изображений, иметь представление о размерах вашей иллюстрации по пустому прямоугольнику. В противном случае вместо изображения будет выведена маленькая пиктограмма и форматирование вашей страницы будет нарушено. Также это увеличивает скорость восприятия информации. Программа просмотра сначала выводит правильно отформатированный текст и, пока читатель просматривает его, загружает изображения в отведенные для них места страницы.

Для указания размеров изображения (в пикселах) служат атрибуты height и width тега <img>.

Если текст слишком быстро подходит к изображению, то возможно использование атрибутов vspace и hspace для указания расстояния (по вертикали и по горизонтали) между кромкой текста и краям иллюстрации.

Для задания фона страницы в теге <body> следующие атрибуты bgcolor= цвет в двойных кавычках, либо background - определяет изображение для заливки фона. Значения задается в виде полного URL или имени файла с картинкой формата gif или jpg.

Выполнить задания №№ 2,3,4,5 из практических работ по НТМL

**25.01.2022г.**

**Тема 1.3 Гипертекстовый документ. (1 ч.)**

**Элемент привязки**  определяет  **гиперссылку (**href**)** на целевую веб - страницу. Каждый элемент привязки начинается тегом <a> и заканчивается тегом </a>. Открывающий и закрывающий теги привязки окружают текст, по которому необходимо выполнить щелчок мышью для перехода по гиперссылке. Чтобы настроить гиперссылку, примените атрибут **href,**  содержащий имя и местоположение файла, к которому предоставляется доступ.

Код элемента привязки следующий:

<ahref= “http://eksmo.ru/”>Издательство «Эксмо» </a> .

Обратите внимание, что значение атрибута href - это URL - адрес веб - сайта. Текст, введенный между двумя тегами <a>, отображается на веб - странице как гиперссылка, и в большинстве случаев подчеркивается браузером. При наведении на гиперссылку указатель мыши превращается в руку с вытянутым указательным пальцем.

**Абсолютная гиперссылка**  указывает абсолютное местоположение ресурса в Интернете. Она применяется, когда необходимо сослаться на ресурсы, расположенные на других веб - сайтах. В значении атрибута href абсолютной гиперссылки на главную страницу веб - сайта содержится протокол http:// и доменное имя.

<ahref=”http://eksmo.ru”> Издательство «Эксмо» </a>

Обратите внимание, чтобы попасть на другие страницы веб - сайта книги, нужно указать название конкретной папки или имя файла. Например, этот элемент привязки создает абсолютную гиперссылку на файл moscow.php, хранящейся в папке purchase\retail\books\ на сайте издательства:

<ahref=”http://www.eksmo.ru/purchase/retail/books/moscow.php”> Издательство «Эксмо», магазины в Москве </a>.

Когда необходимо ссылка на нужную веб- страницу в пределах вашего сайта, используется  **относительная ссылка.** Значение атрибута href относительной гиперссылки не начинается с протокола http:// и в нем не указывается доменное имя. Значение href содержит только имя файла (иногда вместе с именем папки) веб- страницы, которую нужно отобразить. Гиперссылка указывает местоположение файла относительно страницы, отображаемой в данный момент. Например, если на вашем веб- сайте есть главная страница, которая называется index.html, и вы хотите создать ссылку на страницу email.html, и вы хотите создать ссылку на страницу email.html, которая находится в той же папке, что и index.html, HTML - код для относительной ссылки будет следующим:

<a href =”e-mail.html”>Контакты</a>.

Элемент привязки также можно использовать для создания ссылок на адрес электронной почты.  **Ссылка на адрес электронной почты** автоматически запустит программу электронной почты, настроенную по умолчанию на компьютере пользователя. Значение атрибута href ссылки e- mail начинается со значения mailto:, а затем указывается действительный адрес электронной почты.

Например, чтобы создать ссылку на адрес электронной почты [help@terrymorries.net](mailto:help@terrymorries.net), напишите следующий код:

<a href=”mailto:help@terrymorries.net”>[help@terrymorries.net](mailto:help@terrymorries.net)</a>.

Рекомендуется писать адрес электронной почты как на веб - странице, так и внутри элемента привязки. Не на всех компьютерах браузер настроен на работу с почтовой программой.

Помещая почтовый адрес в обеих локациях, вы повышаете юзабилити для всех посетителей.

Чтобы задать цвет для ссылок, для этого в теге <body> указываются следующие атрибуты:

Link- определяет цвет гиперссылок в документе.

ALink- определяет цвет подсветки гиперссылок в момент нажатия.

VLink – определяет цвет гиперссылок на документы, которые вы уже просмотрели.

Значения для атрибутов записываются в виде либо RGB- значением в шестнадцатеричной системе, либо одним из базовых 16 цветов.

Выполнить задания №№ 14,15 из практических работ по НТМL

**25.01.2022г.**

**Тема 1.4 Формы. (1 ч.)**

На многих Web- страницах встречаются элементы пользовательского интерфейса: поля ввода данных, кнопки, переключатели и др. Внешний вид этих элементов обычно соответствует их назначению. Так, в поле ввода данных можно ввести символы посредством клавиатуры, а на кнопку можно навести указатель мыши и щелкнуть ее левой клавишей.

Средства (X)HTML позволяют отправить данные на сервер для того, чтобы серверное программное обеспечение их как-то обработало или, в простейшем случае, просто сохранило на своем, удаленном серверном диске. Для выполнения этой процедуры со стороны клиента (документа, загруженного в браузер) ничего другого, кроме (X)HTML, не требуется.

Для обеспечения пересылки данных, введенных пользователем в соответствующие элементы на web- странице и/или сгенерированных программно с помощью сценариев, достаточно создать так называемую форму. В (X)HTML - документе для этого предусмотрен контейнерный тег <form>, внутри которого можно расположить один или несколько тегов, создающих элементы пользовательского интерфейса – поля ввода данных, переключатели и кнопки. При соблюдении некоторых простых правил создания формы данные, находящиеся в ее элементах, можно передать серверу.

С помощью неконтейнерноговнутристрочного тега <input> можно создать элементы различных типов: поля ввода данных, кнопки и переключатели (флажки и радиокнопки). Тип элемента определяется атрибутом type. В зависимости от типа элемента могут использоваться те или иные дополнительные атрибуты. Некоторые типы предусмотрены специально для передачи на сервер данных, ассоциированных с элементами. Атрибут type тега <input> может принимать следующие значения:

* Text- текстовое поле ввода данных; дополнительные атрибуты:
* Maxlength - максимальное число вводимых символов (по умолчанию - не огарничено);
* Size – ширина поля, выраженная в количестве одновременно видимых символов;
* Value- значение (содержимое поля);

**Элементы, заданные тегом <input>**

|  |  |  |  |
| --- | --- | --- | --- |
| Поле ввода данных <inputtype=”text” value= Привет> |  | Кнопка  <input type=”button” value=”Щелкниздесь”/> |  |
| Флажок  <inputtype=”checkbox” value= “ясогласен” checked=”checked”/> |  | Кнопка передачи данных  <inputtype=”submit”/>  <input type=”submit” value=””Отправить> |  |
| Радиокнопка  <input type=”radio” value=”Да”checked=”checked”/> |  | Графическаякнопка  <input type=”image” src=”mybutton.jpg”/> |  |
| Выборфайла  <input type=”file”/> |  | Кнопка отмены изменения данных  <inputtype=”reset”/>  <input type=”reset” value=”Отмена”/> |  |
| Вводпароля  <input type=”password”/> |  |

* Checkbox- переключатель типа флажок; дополнительные атрибуты:
* Value- значение, ассоциированное с флажком;
* Checked=”checked” - если указан, то флажок установлен;
* Radio – селекторный переключатель (радиокнопка); дополнительные атрибуты:
* Value- значение, ассоциированное с переключателем;
* Checked=”checked” - если указан, то переключатель установлен (выбран);
* File- текстовое поле ввода данных с кнопкой **Обзор,** щелчок на который открывает диалоговое окно выбора файла; полное имя выбранного файла отображается в поле ввода. Дополнительные атрибуты maxlength, size и valueтакие же, что и для обычного поля ввода данных, задаваемого атрибутом type=”text”;
* Password – текстовое поле ввода данных, в котором все символы отображаются в виде «звездочки» (\*) или закрашенного кружка. Дополнительные атрибуты maxlength, size и value те же, что и для обычного поля ввода данных, задаваемого атрибутом type=”text”. Обычно в такое поле вводят пароль. Однако, хотя вводимые в поле типа password символы маскируются «звездочками», передача их на сервер не шифруется;
* Button – кнопка; дополнительный атрибут value принимает в качестве значения строку, отображаемую на кнопке. По умолчанию на кнопке нет надписи;
* Submit – кнопка, щелчок на которой приводит к отправке данных, содержащихся в форме, на сервер; дополнительный атрибут value принимает в качестве значения строку, отображаемую на кнопке. По умолчанию на кнопке находится надпись **Подача запроса, Отправить запрос, Начали** или **Submit**в зависимости от браузера. Передача данных произойдет только при определенных условиях.
* Image- графическое изображение, играющее такую же роль, что и кнопка типа submit; адрес графического файла указывается как значение атрибута src;
* Reset – кнопка, щелчок на которой отменяет все изменения в элементах, содержащихся в форме (восстанавливает исходные значения); дополнительный атрибут value принимает в качестве значения строку, отображаемую на кнопке. По умолчанию на кнопке находится надпись **Сброс, Сбросить** или **Reset** в зависимости от браузера;
* Hidden- невидимый элемент; пользователь не может изменить данные, сохраняемые как значение атрибута value. Если этот элемент находится в форме, то его данные можно передать на сервер. Данные (значения атрибута value) невидимого элемента могут быть сформированы сценарием.

Тег <select> позволяет создать раскрывающийся комбинированный список, из которого можно выбрать элемент. Это контейнерный тег, внутри которого находятся теги <option>, предназначенные для задания элементов (опций) списка.

Тег <select> имеет множество атрибутов, среди которых отметим основные:

* Accesskey- определяет клавишу быстрого доступа к раскрывающемуся списку; значение атрибута- строка, содержащая букву или цифру. Если нажать на указанную клавишу в сочетании с клавишей <Alt>, то элемент получит фокус. Например, <select accesskey=”S”>;
* Name- имя (идентификатор) списка;
* Disabled=”disabled”- делает список недоступным пользователю;
* Multiple=”multiple” – обозначает возможность выбора из списка одновременно нескольких элементов;
* Size - число одновременно видимых элементов списка; по умолчанию-1. Если это число больше единицы, то список снабжается полосой прокрутки;
* Tabindex- целое число, определяющее порядок перехода к элементу с помощью клавиши <TAB>; обычно используется, когда в форме присутствуют несколько элементов интерфейса, а порядок ввода данных в них важен.

Атрибуты тега <option>:

* Selected=”selected” – обозначает выбранный (выделенный) элемент списка;
* Value – значение, ассоциированное с элементом списка.

Текстовое содержимое элемента списка указывается сразу после тега <option>.

Пример раскрывающегося списка:

<html>

<head><title>Раскрывающийся список </title></head>

<body>

<select name=”List1”>

<option value=”1”/>Первый

<option value=”2”/>Второй

<option value=”3”/>Третий

<option value=”4”/>Четвертый

</select>

<br/><br/>

<select name=”List2” size=”3”>

<option value=”Один”/>Один

<option selected=”selected” value=”два”/>Два

<option value=”Три”/>Три

<option value=”четыре”/>Четыре

</select>

<br>

Оценитеэтотсайт:

<select name=”List3” size=4>

<option value=”5”>Отлично

<option selected=”selected” value=”4”>Хорошо

<option value=”3”>Таксебе

<option value=”0”>Плохо

</select>

</body>

</html>

Текстовая область - это прямоугольное поле с полосами прокрутки, внутри которого можно разместить многострочный текст. Его можно сделать как редактируемым, так и доступным только для просмотра. Для создания текстовой области применяют контейнерный внутристрочный тег <textarea>. Важная особенность этого тега в том, что все теги, заключенные в него, не выполняются браузером, а отображаются как обыкновенный текст.

Основные атрибуты тега <textarea>:

* Accesskey- определяет клавишу быстрого доступа к текстовой области; значение атрибута – строка, содержащая букву или цифру. Если нажать указанную клавишу в сочетании с клавишей <Alt>, то элемент получит фокус. Например, <textareaaccesskey=”T”>.
* Cols- число символьных позиций по горизонтали для отображения текста;
* Rows- число строк, одновременно видимых в окне текстовой области;
* Readonly=”readonly” – устанавливает текстовую область в режим просмотра (только для чтения); если этот атрибут опустить, то текстовая область будет работать как поле ввода и редактирования данных;
* Name- имя (идентификатор) текстовой области;
* Disabled=”disabled”-делает область недоступной пользователю;
* Tabindex- целое число, определяющее порядок перехода к элементу с помощью клавиши <TAB>; обычно необходимо, когда в форме присутствует несколько элементов интерфейса, а порядок ввода данных в них важен;
* Wrap – определяет режим автоматического переноса символов на другую строку; возможные значения:physical, virtual и off (выключено). По умолчанию включен автоматический перенос посимвольно.

Пример:

<html>

<head><title>текстоваяобласть</title></head>

<body style=”background:#e0e0e0”>

<textarea cols=15 rows=5>

Привет

</textarea>

<br><br>

<textarea cols=25 rows=8 wrap=off>

<html>

<h1>Примерhtml- кода</h1>

<select name=”List1”>

<option value=”1”>Первый

<option value=”2”>Второй

<option value=”3”>Третий

</select>

</html>

</textarea>

</body>

</html>

Тег <form> используется в качестве контейнера для элементов пользовательского интерфейса (<input>, <select>, <textarea> и <button>), образуя так называемую форму. Благодаря специальным атрибутам этого тега с помощью кнопки типа submit (<inputtype=”submit”>) можно передать на сервер данные, содержащиеся в элементах формы.

Атрибуты тега <form>:

* Action –URL- адрес сервера или конкретного серверного приложения, обрабатывающего переданные ему данные формы. Для передачи данных этот атрибут обязателен: если он не указан или его значение пусто, данные формы не будут отправлены никаким способом.
* Method- метод передачи данных на сервер:get или post
* Enctype- метод кодирования данных, передаваемых на сервер;
* Name- имя формы
* Target- строка, содержащая имя (значение атрибута name) окна или фрейма, в котором следует показать ответ сервера; возможны также и следующие значения:
* \_parent – родительский фрейм, а если фреймов нет, то данный параметр работает как self
* \_blank- новое окно браузера
* \_top- отменяет все фреймы и загружает ответ сервера в главное окно браузера, а если фреймов нет, то данный параметр работает как self
* \_self- загружает ответ сервера в текущее окно или фрейм

Пример:

<html>

<head><title>Форма</title></head>

<body>

<form action=<http://anyserver.ru/anyprogram.php> method=”post”>

<h2>Сообщение автору </h2>

Вашe-mail:

<input type=”text” name=”from” value=”” size=30>

<br>

Сообщение: <br>

<textarea name=’text’ cols=30 row=8></textarea>

<br>

<input type=”reset”>

<input type=”submit” name=”Отправитель” value=”Отправитель”>

</form>

</body>

</html>

**25.01.2022г.**

**Тема 1.5 Таблицы.**

Таблицы в HTMLорганизуются как набор столбцов и строк. Ячейки таблицы могут содержать любые HTML- такие как заголовки, списки, абзацы, фигуры, графика, а также элементы форм и другие таблицы.

**Таблица:<table>….. </table>**

Это основные теги, описывающие таблицу. Все элементы таблицы должны находиться внутри этих двух тэгов. По умолчанию таблица не имеет обрамления и разделителей. Обрамление добавляется атрибутом BORDER.

**Строка таблицы: <tr>….</tr>**

Количество строк таблицы определяется количеством встречающихся пар тэгов <tr>….</tr>. Строки могут иметь атрибуты align и valign, которые описывают визуальное положение содержимого строк в таблице по горизонтали и вертикали соответственно.

**Ячейка таблицы: <td>……</td>**

Описывает стандартную ячейку таблицы. Ячейка таблицы может быть описана только внутри строки таблицы. Если в строке отсутствует одна или несколько ячеек для некоторых колонок, то браузер отображает пустую ячейку. Расположение данных в ячейке по умолчанию определяется атрибутами align=left и valign= middle. Данное расположение может быть исправлено как на уровне описания строки, так и на уровне описания ячейки.

**Заголовок таблицы: <th>……</th>**

Ячейка заголовка таблицы имеет ширину всей таблицы. Текст в данной ячейке выравнен по центру и отображается жирным шрифтом.

**Подпись: <caption>…… </caption>**

Данный тег описывает название таблицы (подпись). Тэг <caption> должен присутствовать внутри <table>…..</table>., но снаружи описания какой – либо строки или ячейки. По умолчанию <caption> имеет атрибут align=top, но может быть явно установлен в align=bottom. Align определяет, где – сверху или снизу таблицы - будет поставлена подпись. Подпись всегда центрирована в рамках ширины таблицы.

**Основные атрибуты таблицы:**

Border= число

Атрибут используется в теге table. Если данный атрибут присутствует, граница таблицы прорисовывается для всех ячеек и для таблицы в целом. BORDER может принимать числовое значение в пикселях, определяющее ширину границы, однако ширина внутренних рамок всегда равна 1, даже если толщина границы 10. Поэтому используют обычно два значения:0 или 1. В первом случае рамки нет, таблица не видна и может служить каркасом страницы, во втором – рамка видна, как у обычных таблиц.

Align= left/right/center/justify

Если атрибут align присутствует внутри тегов <caption>…..</caption>, то он определяет положение подписи для таблицы (сверху или снизу). По умолчанию align=top. Если атрибут align встречается внутри <tr>, <th> или <td>, он управляет положением данных в ячейках по горизонтали. Может принимать значения left(слева), right(справа), center(по центру) или justify (по формату).

Valign= число

Данный атрибут встречается внутри тегов <tr>, <th> и <td>. Он определяет вертикальное размещение данных в ячейках. Может принимать значения top (вверху), bottom(внизу), middle(посередине) и baseline (центр симметрии совпадает с базовой линией шрифта).

Nowrap

Данный атрибут говорит о том, что текст в ячейке не может логически разбиваться на несколько строк и должен быть представлен одной строкой.

Colspan=число

Указывает, какое количество ячеек будет объединено по горизонтали для указанной ячейки. По умолчанию – 1.

Rowspan = число.

Указывает, какое количество ячеек будет объединено по вертикали для указанной ячейки. По умолчанию – 1.

Colspec

Данный параметр позволяет задавать фиксированную ширину колонок либо в символах, либо в процентах, например colspec=”20%”.

Background=URL

Задает фоновый рисунок в таблице

BGColor=color

Цвет фона таблицы

Bordercolor=color

Задает цвет рамки

Пример таблицы с рамками и без рамок:

<html>

<head>

<title>таблицасрамками</table>

</head>

<body>

<table border>

<tr><th> fruits </th><th> vegetables </th><th> whole grains </th></tr>

<tr><td> apple </td><td> broccoli </td><td> barley </td></tr>

<tr><td> orange </td><td> cauliflower </td><td>weat berries </td></tr>

<tr><td> kiwi </td><td> sugar snap pea </td><td> millet </td></tr>

<tr><td> pineapple </td><td> bell pepper </td><td> quinoa </td></tr>

</table>

<hr>

<table>

<tr><th> fruits </th><th> vegetables </th><th> whole grains </th></tr>

<tr><td> apple </td><td> broccoli </td><td> barley </td></tr>

<tr><td> orange </td><td> cauliflower </td><td>weat berries </td></tr>

<tr><td> kiwi </td><td> sugar snap pea </td><td> millet </td></tr>

<tr><td> pineapple </td><td> bell pepper </td><td> quinoa </td></tr>

</table>

</body>

</html>

Смежные ячейки таблицы могут объединяться с целью размещения большего количества данных. Например, в таблице из пяти строк и пяти столбцов все ячейки первой строки можно объединить и поместить в этой строке красивый заголовок таблицы. Возможно также объединение нескольких строк или создание пустой прямоугольной области.

Для соединения двух смежных ячеек в одном столбце нужно использовать атрибут rowspanтега<th> или <td>, например:

<tdrowspan=2>

Для объединения двух смежных ячеек в одной строке нужно использовать атрибут colspan тех же тегов, например:

<td colspan=2>

Пример:

<html>

<head>

<title> Row and Column Spanning </title>

</head>

<body>

<table border>

<tr><thcolspan=3> DC nationals </th><tr>

<tr><th> offense </th><th>defense </th><th> goalie </th></tr>

<tr>

<td>husmann</td><td>O’Donnell </td>

<td rowspan=5>

Weinberg <br> Weinberg <br> Weinberg <br>

Weinberg <br> Weinberg <br> Weinberg <br>

</td>

</tr>

<tr>

<td colspan=2>popplewell</td>

</tr>

<tr>

<td>McGilly</td><td>longo</td>

</tr>

<tr>

<td> Donahue </td><td> Seymour </td>

</tr>

<tr>

<td>camillo</td><td colspan=2>Walsh Walsh Walsh</td>

</tr>

</table>

</body>

</html>

Выполнить задания № 13 из практических работ по НТМL