

Практическая работа №1

Для начала необходимо определиться с основными вещами - местом хранения проекта и ПО для работы. В случае работы на своем компьютере вариаций того и другого очень много, аудиторно - значительно меньше в силу ограничений учебного заведения. Начинаем подготавливать рабочую область

Задание

1. Откройте “Этот компьютер” – “Личный диск”
2. Создайте папку “Информатика” – “Практические работы” – “HTML” – “Практическая работа №1”
3. В зависимости от версии ОС необходимо сделать видимыми расширения типов файлов. Либо через вкладки “Параметры” – “Вид” – “Скрывать расширения для зарегистрированных типов файлов”, либо то же самое, но через “Панель управления”-”Параметры проводника”
4. В папке необходимо создать первый файл - текстовый документ с именем index, расширение смените с .txt на .html
5. Теперь необходимо определиться с редактором кода. В меню пуск/на рабочем столе необходимо найти ярлыки возможных редакторов (Notepad++, Visual Studio, etc). Если не повезло - работать придется в программе, присутствующей на рабочей машине в 100% случаев - Блокнот
6. Откройте файл index.html с помощью редактора кода. ПКМ – “Открыть с помощью...”

Приступаем к базовому оформлению

Каждый HTML-документ начинается с декларации типа документа, или «доктайпа». Тип документа необходим, чтобы браузер мог определить версию HTML и правильно отобразить страницу. Для старой версии HTML доктайп выглядел так:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

А для современной версии HTML уже намного проще:

```
<!DOCTYPE html>
```

Данная запись должна идти перед всеми возможными тегами и начинать HTML-документ. Простейшая HTML-страница состоит как минимум из трёх тегов: `<html>`, `<head>` и `<body>`. Тег `<html>` располагается в документе сразу после доктайпа и содержит все остальные теги, являясь для них родительским. Тегу обычно добавляют важный атрибут `lang` (сокращение от английского «language»), в котором задаётся язык документа. Наш сайт будет русскоязычным, поэтому в `lang` пропишем такое значение: `<html lang="ru">`. Тег `<head>` хранит важную служебную информацию, а в теге `<body>` хранится содержимое страницы, которое отображается в окне браузера. Все тексты, картинки и в целом - контентную часть мы будем добавлять внутрь `<body>`. В документе теги `<head>` и `<body>` могут быть использованы только один раз.

Задание

1. На первой строке декларируйте тип документа
2. Далее задайте явную структуру документа с помощью ПАРНЫХ тегов `<html>`, `<head>` и `<body>`, в котором `<html>` является родителем для пары оставшихся
3. Вернитесь в папку, и выполните по файлу ПКМ – «Открыть с помощью…» – Google Chrome (адресация по умолчанию C:\Program Files\Google\Chrome\Application\chrome.exe)

Работаем с “головой”

Итак, первая веб-страница вами уже написана. Мы пока не будем опешлять наше обучение стандартным “Hello, world!”, а начнем работу с основными тегами, необходимыми при создании веб-страницы.

Начать следует с тега `<head>`, который предназначен для хранения служебной информации о странице. Он располагается первым в теге `<html>`, сразу перед `<body>`. Внутри обычно содержится заголовок страницы - тег `<title>`, ключевые слова и описание страницы - группа тегов `<meta>` с разными атрибутами, и другие служебные данные. Также внутри него подключаются внешние ресурсы, например, стили. Содержимое этого тега не отображается на странице напрямую.

Задание

1. Зададим явную кодировку нашей страницы, для этого первой строкой вставим одиночный тег `<meta>` с атрибутом `charset` и значением `utf-8`. Строка должна выглядеть так: `<meta charset="utf-8">`
2. На следующей строке снова обратимся к тегу `<meta>`, но теперь с атрибутом `name` и значением `"keywords"`. Тег будет отвечать за “ключевые слова” для поисковых машин. Добавим тегу еще один атрибут `content`, в котором перечислим подходящие – “вёрстка, HTML, CSS, обучение”
3. Снова `<meta>`, атрибут `name="description"`, а атрибут `content="Мой первый сайт"`. Таким образом мы добавим страницы описательную составляющую страницы. При публикации сайта в сети и попадании его в выдачу поисковика - это как раз тот блок, который находится под найденной ссылкой на сайт
4. Наконец добавим заголовок. Для этого используем парный тег `<title>`, контентной его частью будет Сайт ученика группы 10.103К Иванова И.И. (подставьте свой номер группы и ФИО).

Ближе к “телу”

Тема тега `<meta>` раскрыта далеко не полностью, и имеет другие возможные варианты атрибутов. На данный момент нам вполне достаточно имеющихся. Так как стандартная управляющая информация уже описана в теге `<head>`, пора приступить к контентному наполнению страницы `<body>`, согласно семантическим нормам.

Само тело так же разбивается чаще всего на смысловые блоки:

1. HTML-элемент `<header>`, представляет собой вводный контент, обычно группу вводных или навигационных средств. Он может содержать другие элементы - чаще всего заголовки, а также логотип, форму поиска, имя автора и другие элементы.
2. HTML-элемент `<main>` предназначен для основного контента (содержимого) `<body>` документа (страницы). Основной контент состоит из контента, который непосредственно относится к главной теме документа или её развивает.
3. HTML-элемент `<footer>` представляет собой нижний колонтитул (футер, подвал). Футер обычно содержит информацию об авторе раздела, информацию об авторском праве или ссылки на связанные документы, реже навигацию по сайту.

Задание

1. Создайте 3 равноправных “ребенка” для `<body>` - теги `<header>`, `<main>` и `<footer>`. Все они парные

Заголовки

Теперь нам необходимо озаглавить нашу страницу. Для этого существуют такие HTML-элементы, как `<h1>` – `<h6>`, представляющие собой 6 уровней заголовков секций. `<h1>` это наибольший заголовок и `<h6>` - наименьший.

Примечания к использованию

- Информация о заголовке может использоваться пользовательскими агентами, например, для автоматического создания оглавления для документа, в случае если вы забыли упомянуть тег `<title>` в `<head>`

- Не используйте низкие уровни чтобы снизить размер шрифта: для такого рода видоизменений существует CSS

- Избегайте пропуск уровней заголовков: всегда начинайте с `<h1>`, потом используйте `<h2>`, и так далее.

- Избегайте использования `<h1>` более раза на страницу, дабы не быть проклятыми SEO-шниками и проигнорированными поисковыми роботами

Важность тэгов `h1-h6` заключается в том, что они создают контуры документа. Контуры документа — это что-то вроде оглавления веб-сайта. Они автоматически создаются из разметки на любой данной веб-странице.

До HTML5 контуры документа создавались с помощью тегов заголовков `h1` до `h6`. Каждое использование заголовка означало бы начало нового раздела контента.

Задание

1. Внутри `<header>` создайте заголовок первого уровня `<h1>` с тем же контентом, что и у тега `<title>`
2. Сохраните документ¹, переключитесь на браузер и обновите страницу

¹ И в целом делайте это почаще

Навигация

Продолжаем наполнение тега `<header>`. Следующий на очереди стать его ребенком - тег `<nav>`. HTML-элемент `<nav>` определяет отдельную секцию документа, назначение которой обозначение ссылок навигации (как внутри текущего документа, так и ведущих на другую страницу). В качестве примера такой секции можно привести меню, якорные ссылки. Детально о теге:

- Не обязательно все ссылки должны быть обернуты в `<nav>`. `<nav>` следует использовать лишь для главных навигационных блоков. Например, `<footer>` часто содержит список ссылок, которые не стоит оборачивать в `<nav>`.
- Документ может содержать несколько `<nav>` элементов. Например, один для навигации по сайту, второй для навигации по странице.
- Пользовательские агенты, такие как устройства чтения с экрана, предназначенные для людей с плохим зрением, могут использовать этот элемент, чтобы определить следует ли пускать рендеринг содержимого навигации.

Сам по себе - тег не имеет своего вида - как и любой `<div>`, и формирует область веб-документа для скринридеров, упрощая навигацию по странице поисковикам и людям со слабым зрением. Поэтому его контентную часть мы будем наполнять одним из самых важных элементов интернет приложений - ссылками. Для этого используется тег `<a>` (сокращение от «anchor»). HTML-элемент `<a>` определяет гиперссылку для перехода на определённое место на странице или на другую страницу в Интернете. Так тег имеет крайне важное значение - мы сразу затронем работу с его атрибутами. Всего тег насчитывает около двух десятков атрибутов, но часть из них устарела, часть пока поддерживается не всеми устройствами, поэтому мы изучим самые известные:

1. `href` - самый важный и необходимый атрибут. Атрибут `href` указывает ссылку: либо URL, либо якорь. Якорь — это название после символа `#`, который указывает на элемент (ID) на текущей странице. URL не ограничены только ссылками на HTTP, они могут использовать любой протокол, поддерживаемый браузером. Например, `file`, `ftp` и `mailto` работают в большинстве браузеров.

2. `download` - Этот атрибут сообщает о том, что эта ссылка должна быть использована для скачивания файла, и, когда пользователь нажимает на ссылку, ему будет предложено сохранить файл как локальный. Если у этого атрибута есть значение, оно будет использовано как заполненное название файла в Окне сохранения, которое открывается, когда пользователь нажимает на ссылку (пользователь может поменять название перед сохранением файла). Ограничений на позволенные значения нет (хотя оно будет конвертировано в нижние подчёркивания, предотвращая специфичные пути), но стоит обратить внимание, что у большинства файловых систем есть ограничения на то, какие знаки препинания поддерживаются файловой системой, и браузеры регулируют названия согласно ограничениям.

3. `target` - Этот атрибут определяет, где показать содержимое по ссылке. Значение по умолчанию `_self` — загружает документ в текущем фрейме. Варианты значений:

- `_blank` загружает документ в новой вкладке.
- `_parent` загружает документ в родительской вкладке. Если нет родителя, параметр будет вести себя как `_self`
- `_top` загружает в окне высшего уровня

Тег `<a>` является парным, поэтому стоит упомянуть — что ссылкой может являться абсолютно любой HTML-элемент, обернутый в него

Задание

1. Создайте семантический узел навигации с помощью парного тега `<nav>`
2. Внутри него начнем располагать ссылки, первой будет ссылка `<a>` на сайт колледжа, атрибуту `href` присвойте значение "<https://natk.ru/>", в контентной части достаточно поставить просто `NATK <a>`
3. Вторая - будет ссылаться на файл "form.html", контентная часть – Анкета. Так же, пока добавим атрибут `download`
4. Третья кнопка навигации получит ссылку на все тот же сайт колледжа, приобретя атрибут `target="_blank"`. А вот с контентной частью чуть интереснее - откройте Paint и создайте изображение кнопки своей мечты. размерами 200 на 50 пикселей. Файл назовите `button.png` и сохраните в ту же папку, что и `index.html`
5. Для подключения изображения используйте тег ``, а конкретнее следующую запись - ``

Наполняем жизнью тег <main>

Содержимое элемента <main> должно быть уникальным для документа. Содержимое, которое повторяется в наборе документов или разделах документа, такое как боковые панели, навигационные ссылки, информация об авторских правах, логотипы сайта и поисковые формы, не должно добавляться, за исключением формы поиска, если она является основной функцией страницы.

Опять таки - данный тег является якорем лишь для скринридеров и поисковых роботов, поэтому сам по себе не имеет отображения. Мы наполним его двумя тегами <section>. Парный HTML-элемент <section> представляет собой автономный раздел — который не может быть представлен более точным по семантике элементом — внутри HTML-документа. Как правило, но не всегда, разделы имеют собственный заголовок. В первый мы вставим новый тег - тег <p>. HTML-элемент <p> представляет собой абзац. Абзацы обычно представлены в визуальной среде в виде блоков текста, отделенных от соседних блоков пустыми строками и/или отступом в первой строке, кроме этого HTML-абзацы могут представлять собой структурную группировку однотипного содержимого, например изображений или полей формы.

Задание

1. Внутри тега <main> создайте два парных тега <section>
2. Внутри первого создайте заголовок второго уровня(помним что заголовок первого уровня у нас уже есть), в нем запишите “О себе”
3. После создайте элемент-параграф, контентная часть которого должна по меньшей мере 100 символов, например “Всем привет! Добро пожаловать на мой первый сайт. Здесь мы запишем текст параграфа, о том как же невероятно интересно изучать основы веба”
4. Вторую секцию обозначьте заголовком “Мои навыки”

Работа со списками

Тут все так же просто, как и раньше. Существует по меньшей мере 2 вида списков - упорядоченный и нет (также известен как *bullet list*). Чтобы обозначить границы списка используются соответствующие для них теги - `` (*unordered list*) - неупорядоченный список и `` (*ordered list*) - упорядоченный список. Атрибуты для упорядоченного списка:

1. `reversed` - Атрибут логического значения (*bool*) показывает, что предметы указаны по списку в обратном порядке. Пункты в списке будут пронумерованы от большего к меньшему.
2. `start` - Нумерация начнётся с указанного числа. Арабскими цифрами (1, 2, 3, и т.д.), даже когда нумерация `type` в буквах или Римском исчислении. Например, чтобы начать нумерацию с буквы "г" или Римской "iv", используйте `start="4"`.
3. `type` - Задаёт тип нумерации:
 - a. a для строчных букв
 - b. A для заглавных букв
 - c. i для строчной Римской нумерации
 - d. I для заглавной Римской нумерации
 - e. 1 для цифр (по умолчанию) указанный тип используется для всего списка, если только не указан любой другой атрибут `type` в элементе ``.

Неупорядоченный список на данный момент не имеет не устаревших атрибутов, а весь его внешний вид регламентируется CSS-свойствами и псевдо-элементами (тема отдельных практических работ). В целом, элементы `` и `` используются для списков и различие лишь в том, что в элементе `` порядок имеет значение. Согласно эмпирической закономерности (или правилу большого пальца²), чтобы определить, какую маркировку использовать, попробуйте поменять порядок элементов в списке. Если суть списка меняется, то тогда подойдёт элемент ``, в противном случае используйте ``.

Внутреннее наполнение и того и другого списков - наполняется уже упомянутым тегом `` - *list item* - единицей списка. Данный тег будет добавлять маркер вокруг контентной части единицы нашего списка

Задание

1. Создайте упорядоченный список, в атрибутах примените стартовое значение, например 5, и тип нумерации, например строчные буквы
2. Внутри тега создайте 5-7 единиц списка с вашими навыками, например "знание HTML" (вы ведь уже начали работу над этим), "знание CSS", "знание JS" и т.д.
3. Переключитесь в браузер и оцените проделанную работу

² «Правило большого пальца» (*rule of thumb* (англ.)) - термин, основанный на городской легенде, согласно которому якобы существовавший английский закон XVII века устанавливал максимальный диаметр палки, которой муж имеет право бить жену, в толщину большого пальца его руки

Завершаем работу со страницей

HTML-элемент `<footer>` представляет собой нижний колонтитул (футер, подвал) для своего ближайшего секционного контента. Футер обычно содержит информацию об авторе раздела, информацию об авторском праве или ссылки на связанные документы.

Задание

1. Создайте элемент футер
2. В контентную часть впишите фамилию и инициалы, а перед ним можно добавить ASCII-символ. Для этого зажмите ALT и наберите на нумпате 0169
3. Обязательно сохраните результат вашей работы
4. Составьте конспект об основных сведениях о тегах, использованных в процессе работы