

Проектирование и разработка корпоративных информационных систем

09.04.03 Прикладная информатика направленность бизнес-модели и цифровые решения

<http://vikchas.ru>

Тема: ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ СИСТЕМ В ЦИФРОВОЙ ЭКОНОМИКЕ

Лекция 2



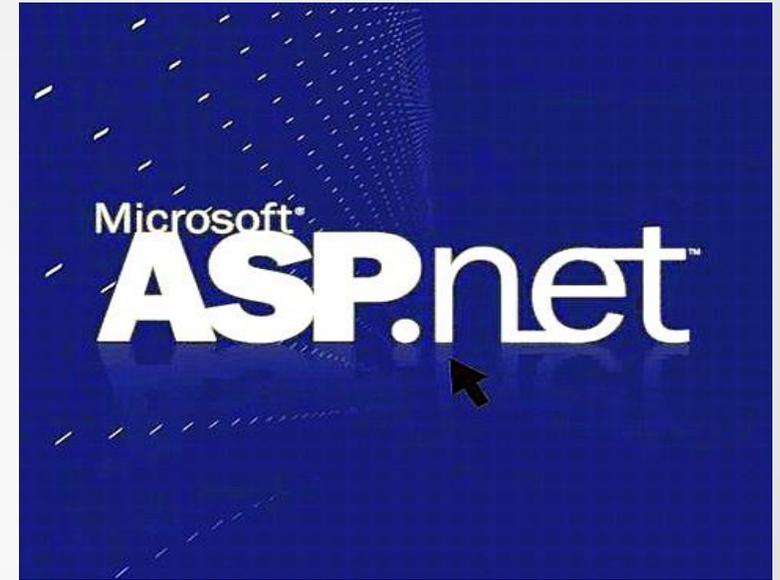
Часовских Виктор Петрович

д-р техн. наук, профессор кафедры ШИиКМ

ФГБОУ ВО «Уральский государственный экономический университет»

Екатеринбург 2021

Веб технология ASP.NET (Active Server Pages) это платформа для создания интерактивных веб приложений, от простого сайта визитки до огромных порталов в полном соответствии с веб стандартами. Она поддерживает работу с несколькими языками программирования, входящими в сборку фреймворка: Basic NET, C#, J# и ряд прочих.



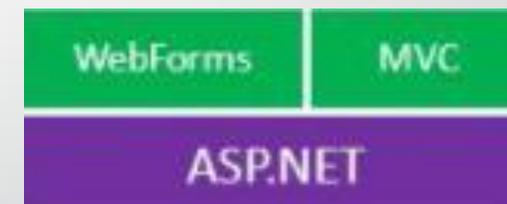
История ASP.NET

1996 – ASP – Active Server Pages, построение страниц на сервере на основе шаблонов. Шаблоны сочетали код на VB с HTML- разметкой.

2001– ASP.NET – Составная часть новой платформы.NET. Технология WebForms по аналогии с WinForms.

2009– ASP.NET MVC. Аналогична уже существующим на рынке подходам: Java Spring 2002, Python Jango 2003 и др

2013– ASP.NET MVC 5.0 – октябрь, последняя версия 2
ASP.NET WebForms ASP.NET WebFormsMVC ASP



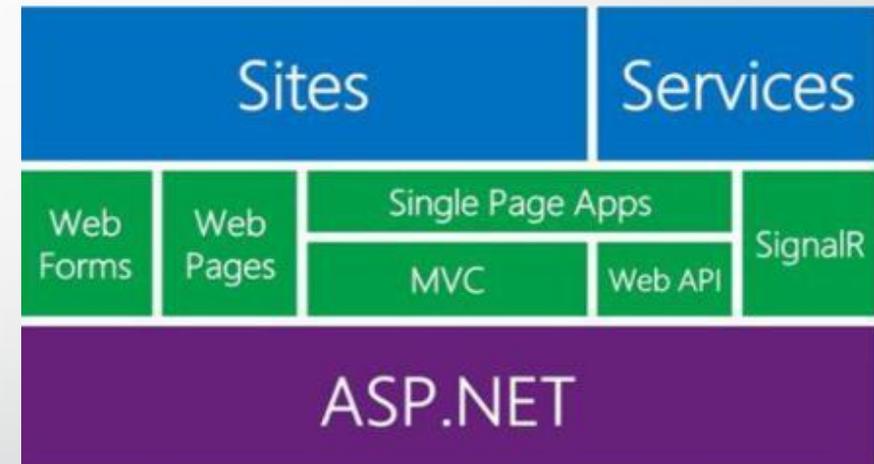
Фреймворки на базе ASP.NET

ASP.NET – бесплатный фреймворк для построения больших веб-приложений с использованием HTML, CSS и JavaScript.

WebForms – технология построение веб-приложений из стандартных управляющих элементов и обработчиков событий.

ASP.NET MVC – построение веб-приложений на базе шаблона MVC с разделением ответственности и полным контролем над HTML кодом страниц.

Web Pages – быстрая разработка веб-сайтов согласно современным веб-стандартам.



Инструменты программирования для ASP.NET

Одно из главных преимуществ *ASP.NET* в сравнении с другими языками и платформами создания веб приложений — это бесплатная доступность полноценных инструментов программирования.

Microsoft Visual Studio Community - бесплатная, полнофункциональная и расширяемая интегрированная среда разработки для создания современных приложений для Windows, Android и iOS, а также веб-приложений и облачных служб. ПО Visual Studio Community создана для индивидуальных разработчиков, проектов с открытым кодом, научных исследований, образования и небольших групп специалистов. Программирование на C#, Visual Basic, F#, C++, HTML, JavaScript, Python и других языках.

Visual Studio Code – бесплатный редактор кода, построен на открытом исходном коде, поддерживает множество языков программирования, C#, Razor, PHP, HTML, XML, CSS, JavaScript, Sass, Python, Perl, F#, C++ др

Microsoft Visual Studio 2019 Professional - коммерческая среда разработки. Для индивидуальных разработчиков и предприятий. Имеет расширенные возможности отладки, диагностики, тестирования и кроссплатформенной разработки.

Веб сервер IIS

IIS (Internet Information Services) — комплекс служб Интернета от компании Майкрософт. IIS распространяется с операционными системами семейства Windows.

Основным компонентом IIS является веб-сервер, который обслуживает веб сайты, расположенные на локальной машине и для сети интернет.

IIS поддерживает протоколы **HTTP, HTTPS, FTP, POP3, SMTP**.

IIS сервер позволяет использовать для сайтов программирование в любой веб системе: ASP.NET, PHP, Python и др., использование любых баз данных, например, MSSQL, MySQL, PostgreSQL, SQLite.

IIS характеризуется малым потреблением ресурсов процессора, является одним из самых быстрых серверов, и что важно, имеет централизованную поддержку компании Microsoft.



Виды веб проектов

Веб-проект ASP.NET можно создавать как **веб приложение** и как **веб сайт**.

Конечная цель обоих проектов - это рабочий сайт, размещенный на сервере.

В веб-приложении весь программный код, находящийся в различных папках и файлах, компилируется в сборку **.dll**.

При работе с веб-сайтом текстовые файлы с программным кодом размещаются непосредственно на сервере для динамической компиляции во время загрузки веб-страниц.

Рекомендация выбора веб приложения

Проекты веб-приложений желательно
выбирать когда:

Необходимо избежать размещения открытого исходного кода на рабочем сервере.

С помощью компилятора требуется создать единую сборку для всего сайта.

Необходимо выполнять модульные тесты кода, находящегося в файлах классов, связанных со страницами ASP.NET.

Требуется ссылаться на классы, связанные со страницами и пользовательскими элементами управления из изолированных классов.

Требуется управление именем и номером версии сборки, созданной для сайта.

Проекты веб-сайтов являются предпочтительным вариантом выбора когда:

Исходные файлы проекта копируются на сервер.

В один веб-проект необходимо включить как код *C#*, так и код на другом языке *.NET*.

Необходимо открывать веб-сайт в Visual Studio, Visual Studio Code, WebMatrix и обновлять его в режиме реального времени.

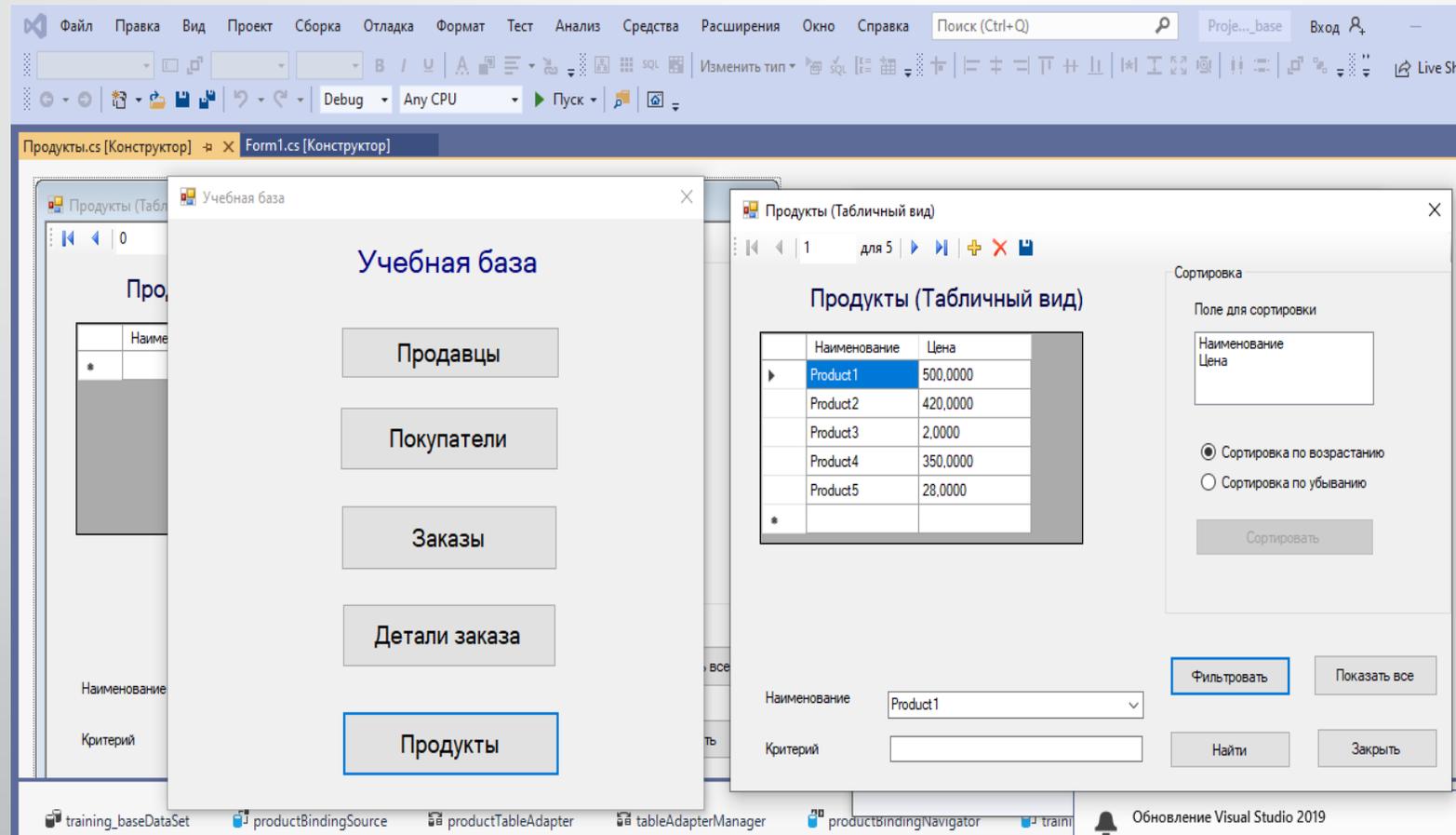
Требуется возможность обновления отдельных файлов в рабочей среде путем простого копирования новых версий на рабочий сервер.

Если требуется создать отдельную сборку для каждой страницы, папки или пользовательского элемента

Вы хотите сохранить исходный код на рабочем сервере в качестве дополнительной резервной копии.

Инструменты работы с Web Forms

Основным инструментом создания и редактирования страниц *Web Forms ASP.NET* является Microsoft Visual Studio .NET. При работе в Visual Studio веб-элементы управления могут добавляться на страницу в ручную или путем перетаскивания из панели инструментов.



HTML

Язык разметки гипертекста (Hyper Text Markup Language) является стандартным средством представления информации в среде Word Wide Web (WWW) в виде веб-страниц.

WWW (World Wide Web) - это часть Internet, представляющая собой гигантский набор документов, хранящихся на компьютерах по всему миру. Сокращенно WWW называют просто Вэб. Веб-страница - это отдельный документ Web, который способен содержать информацию различного вида - текст, рисунки, фото, видео, аудиозаписи.

URL

URL (Uniform Resource Locator) - это уникальный адрес, которым обладает каждая веб-страница в сети. Если пользователю известен URL страницы, то он может ее отобразить у себя в браузере, набрав ее адрес в строке браузера. В общем случае URL состоит из наименования используемого протокола, названия сервера и обозначения пути доступа к странице.

Протокол определяет правила обращения к веб-странице.

Название сервера (доменное имя) - обозначает компьютер, содержащий данную веб-страницу.

Путь доступа - указывает местоположение страницы на диске сервера. Например адрес данной страницы <http://on-line-teaching.com/html/index.html>

`http` - наименование протокола

`on-line-teaching.com` - наименование сервера или доменное имя

`/html/index.html` - путь доступа к странице на самом сервере.

Базовые понятия

Web-страницы - это документы в формате HTML, содержащие текст и специальные тэги (дескрипторы) HTML. По большому счету тэги HTML необходимы для форматирования текста (т.е. придания ему нужного вида), который "понимает" браузер. Документы HTML хранятся в виде файлов с расширением .htm или .html.

Тэги HTML сообщают браузеру информацию о структуре и особенностях форматирования Веб-страницы. Каждый тэг содержит определенную инструкцию и заключается в угловые скобки < >. Большинство тэгов состоят из открывающей и закрывающей частей и воздействуют на текст, заключенный внутри.

Теги бывают одиночными и контейнерными. Контейнером называется пара: открывающий <ТЕГ> и закрывающий </ТЕГ>.

<ТЕГ> Контейнер </ТЕГ>

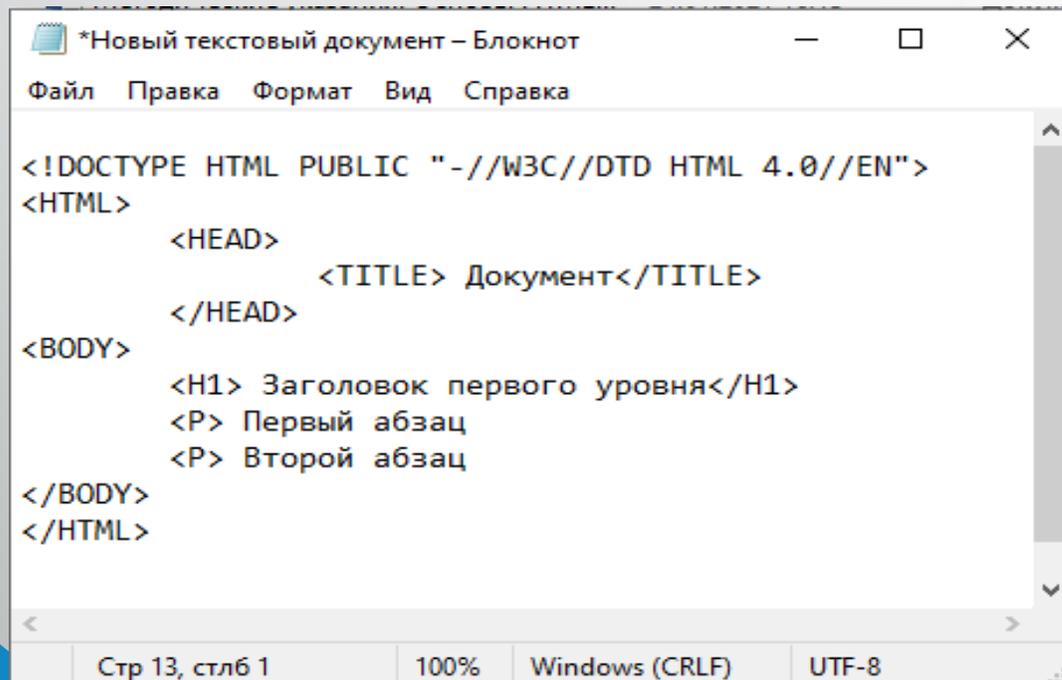
Атрибуты тега задают значения свойств данного объекта или объектов помещенных в контейнер. Значения свойств, содержащие пробелы, берутся в кавычки, в остальных случаях кавычки можно опустить.

Структура HTML –документа

В идеальном случае HTML –документ состоит из трех частей:

- информация о версии используемого HTML;
- заголовок документа;
- тело документа.

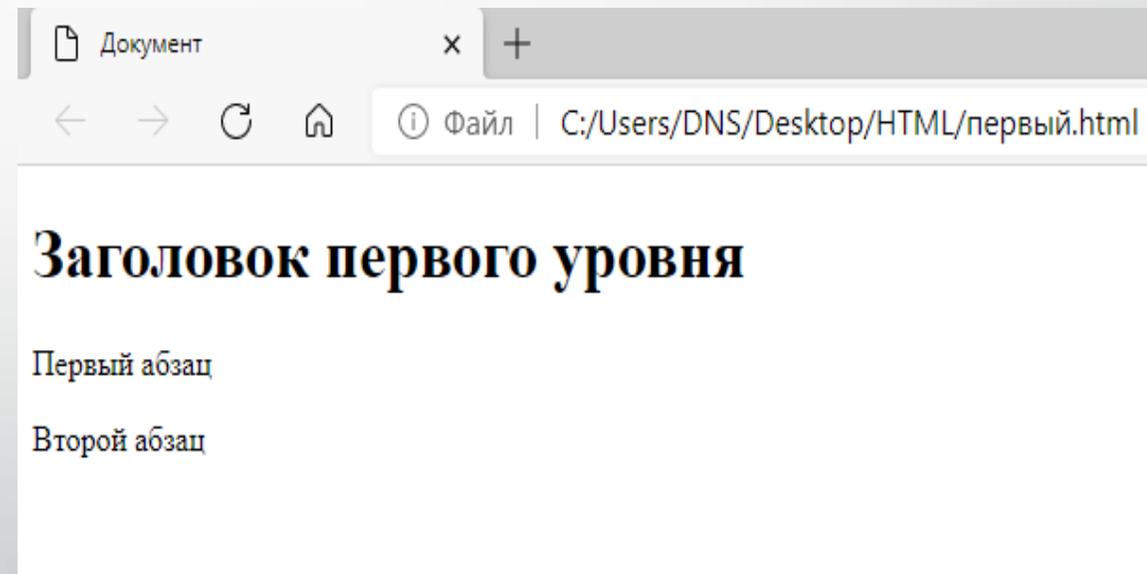
Пример простейшего HTML –документа, содержащий все структурные элементы.



```
*Новый текстовый документ – Блокнот
Файл  Правка  Формат  Вид  Справка

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
  <BODY>
    <H1> Заголовок первого уровня</H1>
    <P> Первый абзац
    <P> Второй абзац
  </BODY>
</HTML>
```

Стр 13, стлб 1 100% Windows (CRLF) UTF-8



Определение тела документа

Наиболее часто используемые атрибуты элемента BODY:

`background` – URL, указывающий расположение изображения для фона (обычно берется небольшое изображение, которое размножается для заполнения фона всего документа);

`bgcolor` – цвет фона HTML-документа; `text` – цвет шрифта документа;

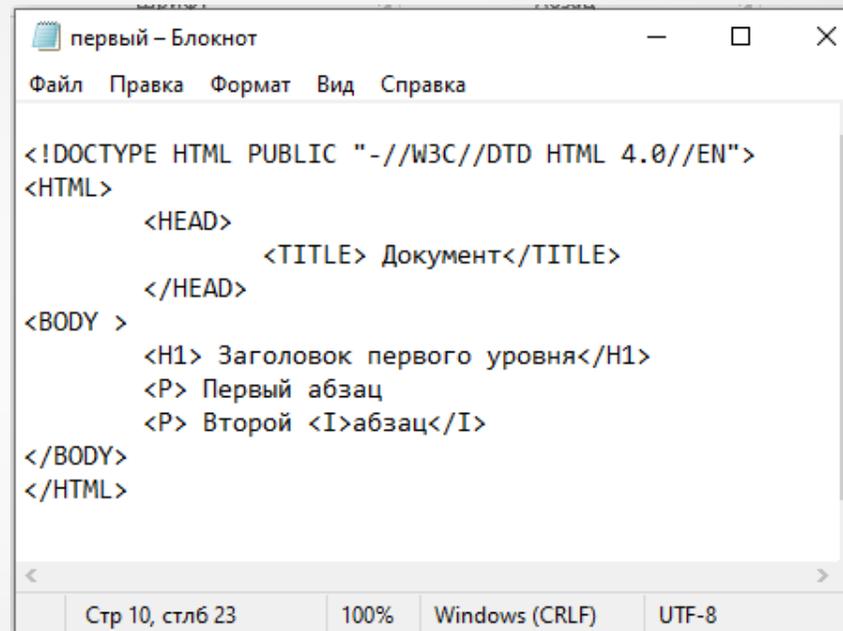
`link` – цвет непосещенных гиперссылок; `vlink` – цвет посещенных гиперссылок;

`alink` – цвет гиперссылок при выборе их пользователем (при нажатии Enter произойдет переход по такой ссылке);

`contenteditable` – позволяет разрешить или запретить пользователю редактирование содержимого HTML-документа при просмотре его браузером (значения `true`, `false`, `inherit`).

Особенности ввода текста

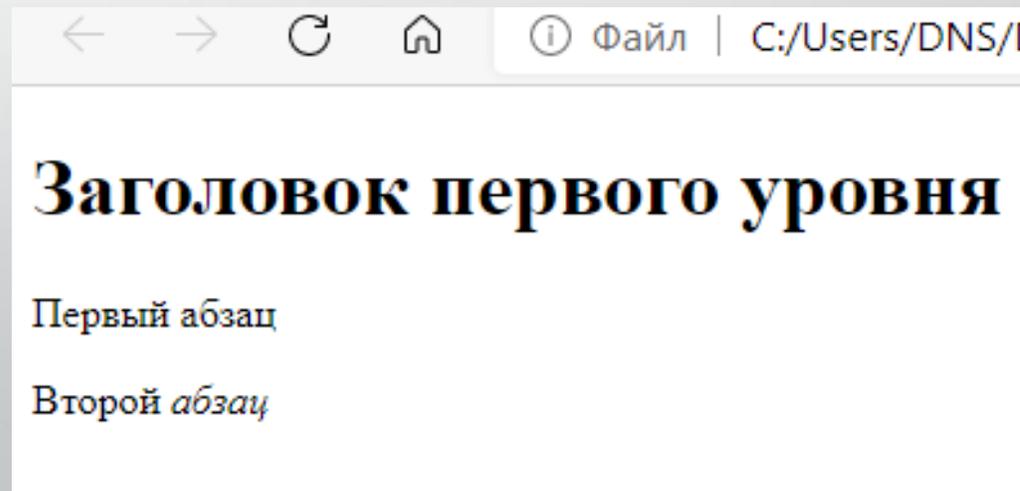
Элемент	Описание
B	Полужирное начертание
I	Курсивное начертание
U	Подчеркнутый текст
STRIKE,S	Перечеркнутый текст
BIG	Текст с увеличенным размером шрифта
SMALL	Текст с уменьшенным размером шрифта
SUP	Верхний индекс
SUB	Нижний индекс
TT	Текст, записанный моноширинным шрифтом (все символы имеют одинаковую ширину)
BLINK	Мерцающий текст (редко поддерживается браузерами).



```
первый - Блокнот
Файл  Правка  Формат  Вид  Справка

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
  <BODY >
    <H1> Заголовок первого уровня</H1>
    <P> Первый абзац
    <P> Второй <I>абзац</I>
  </BODY>
</HTML>

Стр 10, стлб 23    100%    Windows (CRLF)    UTF-8
```



Задание шрифта текста

Если нужно отобразить некоторый текст с использованием определенного шрифта, а не применяемого браузером по умолчанию, то в HTML предусмотрен элемент **FONT**. Он вводится при помощи парных тегов `` и ``.

Параметры шрифта для элемента **FONT** устанавливаются заданием значений следующих его атрибутов:

face — задает название шрифта, например `Arial` или `System`;

size — задает размер шрифта (значение от 1 до 7, по умолчанию используется значение 3);

color — задает цвет шрифта.

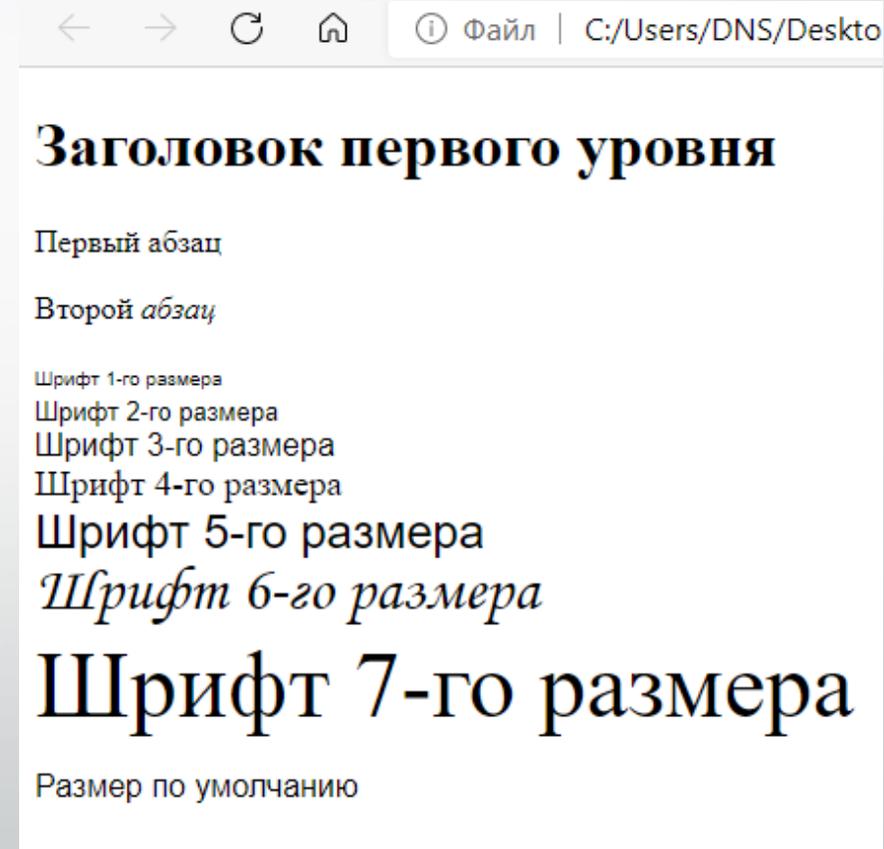
Для атрибута **size** могут использоваться только семь значений.

Задание шрифта текста

```
*первый – Блокнот
Файл  Правка  Формат  Вид  Справка

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
<BODY >
  <H1> Заголовок первого уровня</H1>
  <P> Первый абзац
  <P> Второй <I>абзац</I>

<P>
<Font Size='1' Face='Arial'>Шрифт 1-го размера </Font><BR>
<Font Size='2' Face='Arial'>Шрифт 2-го размера </Font><BR>
<Font Size='3' Face='Arial'>Шрифт 3-го размера </Font><BR>
<Font Size='4' Face='Times New Roman'>Шрифт 4-го размера </Font><BR>
<Font Size='5' Face='Arial'>Шрифт 5-го размера </Font><BR>
<Font Size='6' Face='Monotype Corsiva'>Шрифт 6-го размера </Font><BR>
<Font Size='7' Face='Botang'>Шрифт 7-го размера </Font>
<P><Font Face='Arial'> Размер по умолчанию </Font>
</BODY>
</HTML>
```



← → ↻ 🏠 ⓘ Файл | C:/Users/DNS/Desktop

Заголовок первого уровня

Первый абзац

Второй *абзац*

Шрифт 1-го размера
Шрифт 2-го размера
Шрифт 3-го размера
Шрифт 4-го размера
Шрифт 5-го размера
Шрифт 6-го размера
Шрифт 7-го размера

Размер по умолчанию

Списки

Нумерованные списки

Нумерованные списки применяются для упорядочения приводимых данных. `...` устанавливают начало и конец нумерованного списка, а дескрипторы `...` отмечают отдельные пункты списка. Тип нумерованного списка устанавливается с помощью атрибута `TYPE` дескриптора ``.

`TYPE='1'` 1,2,3,4 ... (задан по умолчанию)

`TYPE='i'` i,ii,iii,iv,...

`TYPE='I'` I,II,III,IV,... `TYPE='a'`
a,b,c,d,...

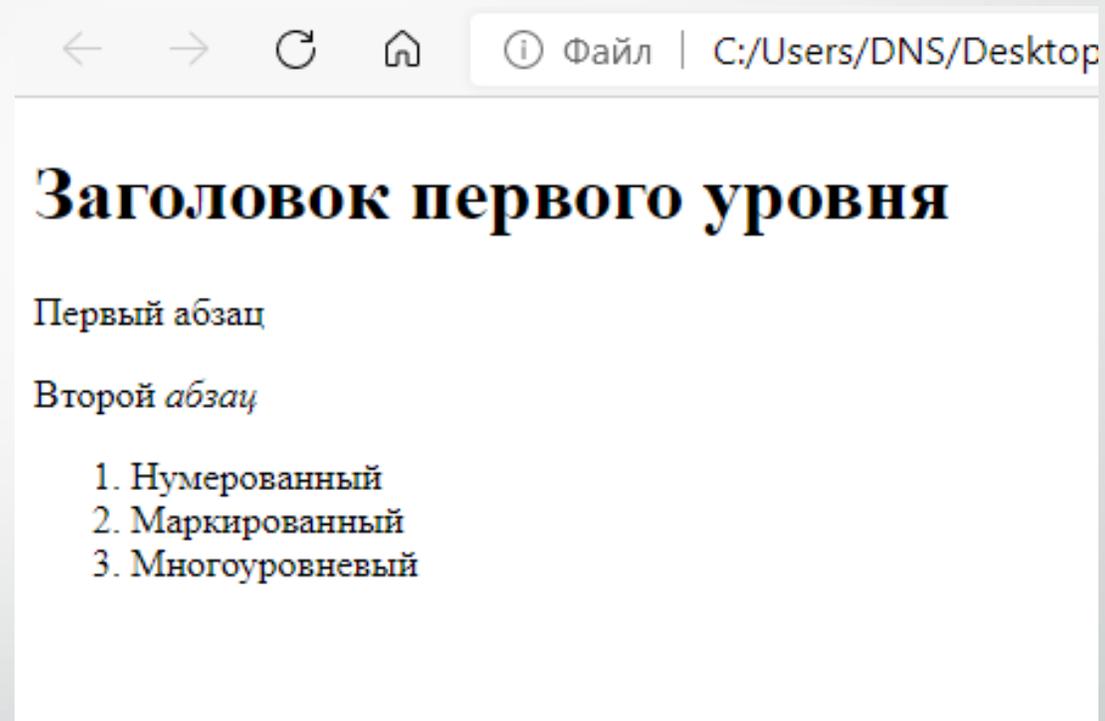
`TYPE='A'` A,B,C,D.

Нумерованные списки

```
*первый – Блокнот
Файл  Правка  Формат  Вид  Справка

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
  <BODY >
    <H1> Заголовок первого уровня</H1>
    <P> Первый абзац
    <P> Второй <I>абзац</I>
  <P>
  <OL>
  <LI> Нумерованный </LI>
  <LI> Маркированный</LI>
  <LI> Многоуровневый</LI>
  </OL>

  </BODY>
</HTML>
```



Маркированные списки

Маркированные списки применяются для перечисления неупорядоченной информации. Для создания маркированного списка используется комбинация двух пар дескрипторов. Дескрипторы `...` устанавливают начало и конец маркированного списка, а дескрипторы `...` отмечают отдельные пункты списка.

Элементы `` и `` имеют ряд атрибутов. Специфичными атрибутами элемента `` являются следующие:

`compact` – заставляет браузер показывать список более компактно.

`type` – позволяет задать тип маркера списка, может принимать значения: `circle` (круг без заливки), `disc` (круг с заливкой) и `square` (квадрат).

Маркированные списки

```
первый - Блокнот
Файл  Правка  Формат  Вид  Справка

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
  <BODY >
    <H1> Заголовок первого уровня</H1>
    <P> Первый абзац
    <P> Второй <I>абзац</I>

    <P>
    <UL TYPE='circle'>
    <LI>Яблоко </LI>
    <LI TYPE='disc'> Груша</LI>
    <LI> Слива</LI>
    <LI TYPE='square'> Вишня </LI>
    <LI> Черешня</LI>
    </UL>

  </BODY>
</HTML>
```

Заголовок первого уровня

Первый абзац

Второй *абзац*

- Яблоко
- Груша
- Слива
- Вишня
- Черешня

Гиперссылки

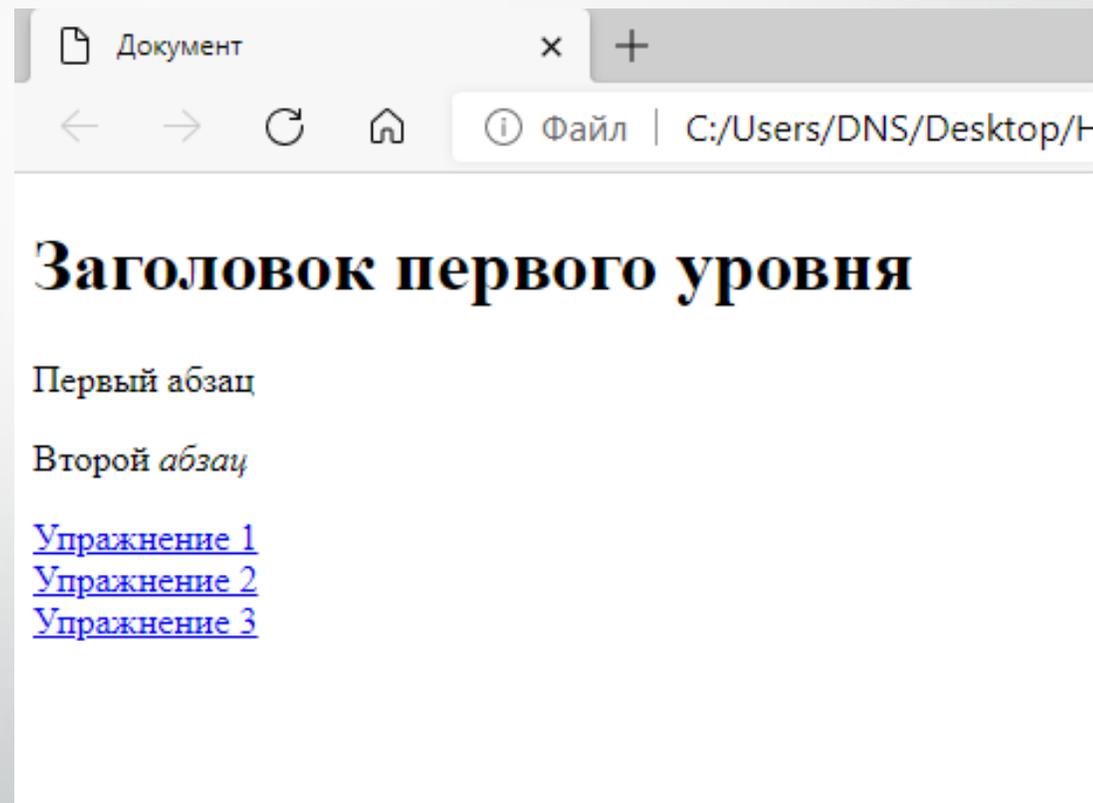
Гиперссылки – ключевой, практически самый главный элемент гипертекста (текста, обладающего возможностями навигации). Гиперссылки позволяют быстро переходить к другим Web - страницам. Гиперссылки исключают необходимость ввода URL необходимой страницы, к которой обращается пользователь. Гиперссылки, как правило, выделяются цветом и подчеркиванием. В качестве гиперссылки может выступать и графическое изображение, в таком случае оно обычно выделяется рамкой.

Гиперссылки

```
первый – Блокнот
Файл  Правка  Формат  Вид  Справка

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
<BODY >
  <H1> Заголовок первого уровня</H1>
  <P> Первый абзац
  <P> Второй <I>абзац</I>
  <P>
  <A href='Упражнение 1.html'>Упражнение 1 </A><BR>
  <A href ='Упражнение 2.html'>Упражнение 2</A><BR>
  <A href ='Упражнение 3.html'>Упражнение 3</A><BR>
</BODY>
</HTML>

Стр 12, стлб 2    100%    Windows (CRLF)    UTF-8
```



Графика на Web – страницах

Для вставки изображения в текст HTML – документа используется элемент одиночный тег `< IMG >`. Список наиболее используемых атрибутов:

`src` – атрибуту `src` присваивается путь к файлу на диске или URL – адрес файла рисунка в Интернете;

`alt` – альтернативный текст, который отображается на месте изображения, если по каким либо причинам само изображение не может быть показано;

`border` – задает толщину границы вокруг рисунка в пикселях;

`align` – определяет выравнивание изображения;

`height` – задает высоту изображения в пикселях;

`width` – определяет ширину изображения в пикселях;

Таблицы

Таблица создается с помощью дескрипторов `<TABLE>...</TABLE>`. Эти дескрипторы создают объект таблицы в том месте текста, где они добавлены в коде HTML. Теперь в таблицу нужно добавить строки и столбцы. Для этого используются дескрипторы:

`<TR>...</TR>` - новая строка таблицы; `<TH>...</TH>` - ячейка заголовка;

`<TD>...</TD>` - обычная ячейка таблицы.

```

<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
<BODY >
  <H1> Заголовок первого уровня</H1>
  <P> Первый абзац
  <P> Второй <I>абзац</I>

<P>
<TABLE align=center border=3 bordercolor=black>
<CAPTION align=top><B>Доходы от продаж за второе полугодие 2007 года</B></CAPTION>
<!--формирование первой строки шапки таблицы-->
<TR align=center >
<TD rowspan=2><B>Филиал\Период </B> <TD colspan=3><B>3квартал </B>
<TD colspan=3><B>4квартал </B>
<!--формирование второй строки шапки таблицы (название месяцев)-->
<TR align=center >
<TD><B>Июль</B> <TD><B>Август</B><TD><B>Сентябрь </B><TD><B>Октябрь</B><TD><B>Ноябрь</B><TD><B>Декабрь</B>
<!--далее следуют строки с данными (первая ячейка каждой строки – название филиала) -->
<TR align=right ><TD align=left >Филиал1<TD>123123<TD> 323233<TD>323453<TD>231423<TD>323212<TD>243673
<TR align=right ><TD align=left >Филиал2<TD>223523<TD> 225243<TD>314423<TD>212445<TD>373812<TD>274673
<TR align=right ><TD align=left > Филиал3<TD>183123<TD> 186834<TD>323453<TD>231423<TD>323212<TD>243673
<TR align=right ><TD align=left > Филиал4<TD>125163<TD> 334343<TD>123553<TD>167423<TD>254412<TD>132367
</TABLE>

```

Таблицы

Заголовок первого уровня

Первый абзац

Второй абзац

Доходы от продаж за второе полугодие 2007 года

Филиал\Период	3квартал			4квартал		
	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
Филиал1	123123	323233	323453	231423	323212	243673
Филиал2	223523	225243	314423	212445	373812	274673
Филиал3	183123	186834	323453	231423	323212	243673
Филиал4	125163	334343	123553	167423	254412	132367

Формы

Назначение форм состоит в том, чтобы дать возможность посетителю управлять содержимым Web – страницы, вызывать на выполнение сценарии, а так же вводить собственные данные и отправлять их на Web – сервер или по адресу электронной почты.

Создание формы

Для вставки формы в HTML –документ используется элемент FORM. Он задается парными тегами < FORM > </ FORM >. Между этими тегами помещаются описания элементов управления формы.

При создании формы используются следующие атрибуты:

`action` – обязательный для каждой формы параметр, URI программы-обработчика данных формы;

`method` – задает способ отправки данных, введенных в форму, может принимать значения `get` (используется по умолчанию) или `post`;

`enctype` – задает тип данных формы, если используется метод отправки `post`; по умолчанию имеет значение `application/x-www-form-urlencoded`; при необходимости передачи файлов используется значение `multipart/form-data`;

`accept-charset` – применяется при передаче файлов, позволяет указать, какие кодировки используются для каждого из файлов (список строковых значений – названий кодировок), по умолчанию устанавливается значение `UNKNOWN` (приложение на сервере должно само определять кодировку);

`accept` – описывает типы файлов (MIME – типы), передаваемые серверу; если этот параметр не использовать, то серверное приложение должно уметь само определять типы передаваемых ему файлов.

Стандартные элементы управления

Элементы управления, которые используются чаще всего: однострочное текстовое поле, поле для ввода пароля, флажки, переключатели, кнопки (как пользовательские, так и выполняющие стандартные действия), поля имен файлов. Все упомянутые элементы управления отображаются браузером Internet Explorer. Для обозначения всех этих элементов управления используется один HTML-элемент – INPUT.

атрибуты:

`type` – принимает строку, задающую тип элемента управления (по умолчанию используется строка `text` и создается соответственно поле для ввода текста).

`name` – используется для задания имени элементу управления (строка, которая помимо идентификации элемента управления добавляется в данные, отсылаемые серверу);

`value` – начальное значение для полей ввода текста и полей для указания имен файлов, также используется как надпись таких элементов управления, как кнопки; необязательно для всех элементов управления, кроме флажков и переключателей;

`checked` – булев атрибут, если он установлен, то флажок или переключатель считается (и отображается браузером)

```

<HTML>
  <HEAD>
    <TITLE> Документ</TITLE>
  </HEAD>
<BODY >
  <H1> Заголовок первого уровня</H1>
  <P> Первый абзац
  <P> Второй <I>абзац</I>

  <P>
  <FORM>
  <!-- Поле ввода --> <BR>Фамилия:
  <BR><INPUT TYPE='text' VALUE='Иванов'><BR> <!-- Группа переключателей -->
  <H2>Группа переключателей</H2> <BR>
  <INPUT TYPE='radio' NAME='color'>Белый<BR>
  <INPUT TYPE='radio' NAME='color'>Красный <BR>
  <INPUT TYPE='radio' NAME='color'>Желтый <BR>
  <INPUT TYPE='radio' = 'color'>Зеленый<BR>

  <!-- Группа флажков --> <H2>Группа флажков</H2>
  <BR><INPUT TYPE= 'checkbox' NAME='fruit'>Груша <BR>
  <INPUT TYPE= 'checkbox' NAME='fruit'>Персик <BR>
  <INPUT TYPE= 'checkbox' NAME='fruit'>Яблоко<BR>

  <!-- Кнопки -->
  <BR><INPUT TYPE= 'button'VALUE='Пишите мне' ONCLICK=window.open('mailto:MyAddress@mail.ru');>
  <BR><INPUT TYPE= 'button'VALUE='Посмотрите мои фотографии' ONCLICK=window.open('http:\\\\www.MySite.org\\MyFoto.htm');>
  </FORM>

</BODY>
</HTML>

```

Заголовок первого уровня

Первый абзац

Второй абзац

Фамилия:

Группа переключателей

- Белый
- Красный
- Желтый
- Зеленый

Группа флажков

- Груша
- Персик
- Яблоко

MVC

В **ASP.NET** применяется традиционная схема MVC.

Аббревиатура *MVC* происходит от слов *Model-View-Controller*.

Model (Модель) - что будем показывать,

View (Вид) - как будем показывать,

Controller (Контроллер) - кто будет управлять.



версии ASP .NET

.NET Core – кроссплатформенная среда выполнения для приложений из веба или консольных программ. Программные продукты, разработанные на ней, могут успешно исполняться на Linux, Windows, MacOS.

Основные плюсы: мультиплатформенность, открытый исходный код.

.NET Framework – среда исполнения, которая предназначена исключительно для Windows. Помогает в разработке десктопных программ под Windows и веб-приложений ASP .NET под IIS.

Ключевые достоинства: огромный ассортимент готовых библиотек, значительно больше возможностей в сравнении с прошлой платформой.

Краткая история MVC

Впервые концепция *MVC* была сформулирована и описана Трюгве Реенскаугом (норвежский ученый в сфере компьютерных наук и заслуженный профессор университета Осло) в 1979 году, работавшим в то время над языком программирования Smalltalk в Херох PARC. Затем, на практике, была реализована версия *MVC* для библиотеки классов Smalltalk-80. Окончательная версия концепции была опубликована лишь в 1988 году в журнале *Journal of Object Technology*. Впоследствии шаблон проектирования *MVC* стал развиваться и конкретизироваться.

Практический смысл MVC

Практическая польза от создания сайтов по концепции *MVC*:

- Комфортность работы над сложной структурой сайта;
- Сосредоточение в одном секторе программирования;
- Возможность многократного использования отдельных частей приложения;
- Возможность использования множества макетов для любой страницы сайта;
- Небольшое количество шаблонов страниц позволяет создавать богатое разнообразное содержание сайта.

Используя шаблон *MVC* изменить дизайн сайта или его логику работы гораздо проще и быстрее. Даже отдельный разработчик, не боясь запутаться в коде, может создавать веб-приложения высокой организованности.

Разновидности проектов:

ASP.NET Core Web Application: применяется для программирования приложений для веба на ASP .NET Core;

ASP.NET Web Application (.NET Framework): этот тип применим для классических проектов: Web Forms, MVC 5, Web Api, основанных на обычном .NET Framework.