

# HTML Первые шаги.





**Hyper**

**Text**

**Markup**

**Language**

**Язык разметки гипертекста**

# Цель урока:

1. Познакомиться со структурой WEB-документа.
2. Изучить основные команды форматирования WEB-документа.
3. Узнать, как создавать простейшие WEB-документы.
4. Получить практические навыки их создания.



# Знакомство с языком HTML

**Гипертекст** – это электронный документ, который содержит гиперссылки на другие документы.

Гипертекстовый документ предназначен для вывода информации на экран компьютера.

**Гипертекстовая технология** – это технология, базирующаяся на использовании гипертекстовых документов. Ее применяют в компьютерных энциклопедиях и учебных программах, прикладных программах для работы со справочной информацией и для организации доступа к информации в WWW, т.е. при работе с WEB-документами.

**WEB-документ** – это текст, написанный на языке HTML или другом, который предназначен для просмотра электронной информации на экране компьютера с помощью программы-браузера.

Так как **WEB-документ** предназначен для просмотра его на компьютере, то желательно, чтобы он помещался целиком на экране. Поэтому WEB-документы называют еще **WEB-страницами**.

Несколько WEB-страниц на одну тему называют WEB-узлом или **WEB-сайтом**.

WEB-сайты создают WEB-дизайнеры.

**WEB-дизайн** – это совокупность правил и рекомендаций, которыми должны руководствоваться авторы, чтобы их сайты были информативными и выглядели привлекательно.

Команды языка HTML называются **теги** и они записываются в **< >**. Большинство тегов – парные **<html>...</html>**

Документ HTML имеет три структурных типа содержимого:

1. **Теги** – команды в **< >**.
2. **Комментарии** – пояснения к документу. Они помогают разобраться в его содержании **<!-- ... -->**.
3. **Текст** – то, что пользователь видит на экране браузера.

Мультимедийные элементы – картинки, звук, видео – не являются частью HTML-документа и хранятся в отдельных файлах. HTML-документ содержит только ссылки на эти файлы в виде тегов.

# теги и атрибуты

Каждый тег состоит из имени тега, за которым может следовать список атрибутов (параметров)

`<i>`

``

Атрибуты отделяются от имени тега и друг от друга пробелами. Значение атрибута пишется после знака равенства. Если значение состоит из одного слова или цифры, то его можно писать без кавычек. Для значений из нескольких слов кавычки обязательны ( “ ” ).

Тег со всеми атрибутами желательно располагать на одной строке.

Для большинства тегов нужен закрывающий тег:

`<i>` `</i>`

# Структура HTML-документа

`<html>`

`<head>`

`</head>`

`<body>`

`<html>`

`<head>`

`<title>` **Мой первый шаг**

`</title>`

`</head>`

`<body>`

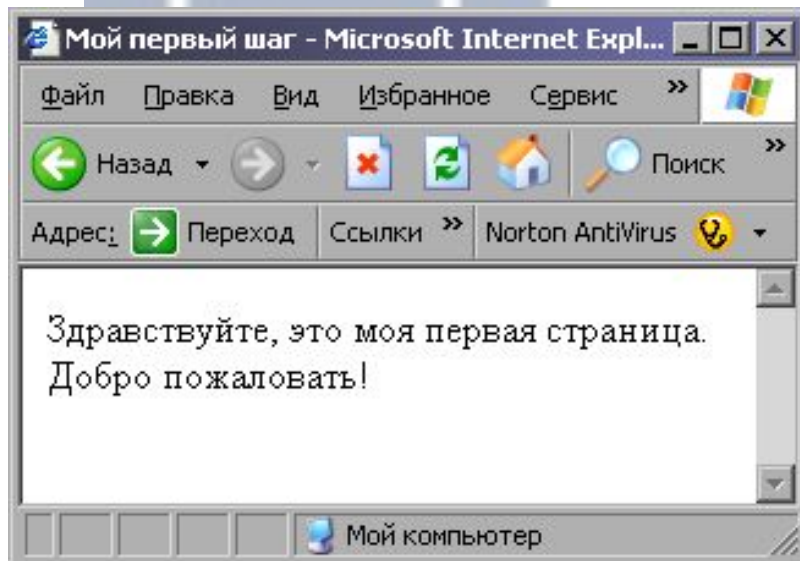
**Здравствуйтесь, это моя  
первая страница.**

`<br>`

**Добро пожаловать!**

`</body>`

`</html>`



`</body>`

`</html>`

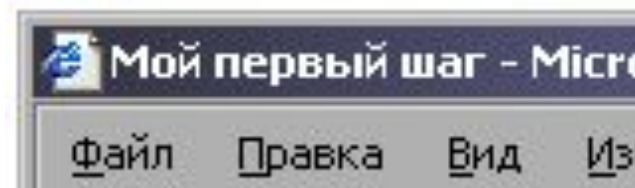


# Заголовок документа – теги <Head> и <Title>

Тег **<Head>...</Head>** включает в себе теги заголовка.

1. Тег **<Title>...</Title>** содержит слова, которые появляются в строке заголовка браузера

```
<html>  
<head>  
<title>Мой первый шаг </title>  
</head>
```



2. Дополнительные теги заголовка:

**<meta>** - содержит дополнительные данные о документе, используемые поисковыми серверами;

**<base>** и **<link>** - определяют базовый адрес документа и некоторые другие

# Тело документа – тег `<body>`

Все, что находится между `<body>` и `</body>`, называется содержимым тела документа.

Тег `<body>` может содержать 3 группы параметров:

1. Управление внешним видом документа.
2. Атрибуты программирования – по событию, таблицы стилей и пр.
3. Атрибуты ссылок и идентификации.

# Параметры тега <body>

**Bgcolor** – изменяет цвет фона. Цвет задается словом или кодом RGB

```
<body bgcolor = "red">
```

```
<body bgcolor = "#FF0000">
```

**Text** – задает цвет текста.

```
<body bgcolor = "red" text="blue">
```

**Background** – помещает в качестве фона изображение из файла с картинкой.

```
<body background = "dog.gif">
```

**Bgproperties** = "fixed" – фоновый рисунок не прокручивается вместе со страницей.

**Link** – задает цвет гиперссылок, по которым пользователь еще «не ходил».

**Vlink** – задает цвет посещенных гиперссылок.

**Alink** – цвет активной гиперссылки (под курсором мышки во время нажатия)

# Оформление текста

## Заголовки

Существует 6 уровней заголовков:

`<h1> ... </h1>`, ...,

`<h6> ... </h6>`

Атрибут – **align** – выравнивание

Значения: **Left** (по умолчанию)

**Right Centr**



```
<body bgcolor = "#CC3399" text = "#CCCCCC">
```

```
<h1 align = "center"> Title 1 </H1>
```

```
<h2 align = "right"> Title 2 </H2>
```

```
<h3 align = "left"> Title 3 </H3>
```

```
<h4> Title 4 </H4>
```

```
<h5> Title 5 </H5>
```

```
<h6> Title 6 </H6>
```

```
</body>
```

# Абзацы

Тег **<p>** указывает на начало нового абзаца и вставляет пустую строку перед абзацем. Атрибут **align**.

Тег **<br>** - разрыв строки. Используется для записи текстов стихов и песен.

Атрибут **clear** позволяет продолжить поток текста после (ниже) рисунка или таблицы. Значения **left, right, all** - продолжают вывод текста там, где указанный в атрибуте край или оба края свободны от таблиц и рисунков. Этот атрибут имеет смысл только с выровненными влево или вправо рисунками или таблицами.

# Точные интервалы

Тег `<nobr>... </nobr>` создает область, в которой текст не разбивается на строки.

Тег `<wbr>` используется в сочетании с `<nobr>` и указывает браузеру место где, в случае необходимости, можно разорвать строку.

Тег `<pre>... </pre>` выделяет часть исходного текста, который должен отображаться «как есть» - с теми же отступами и разбиением на строки. Используется для записи текстов программ. Внутри этого тега могут располагаться гиперссылки, картинки и пр.

Тег `<center>... </center>` - содержимое выравнивается по центру – текст, рисунки, таблицы. Аналог атрибута `align=center`



# Физкультминутка

# Физическое и логическое форматирование текста

1. Физическая разметка документа – это явное указание браузеру, как должен выглядеть тот или иной фрагмент текста – размер шрифта, курсив и т.п.
2. Логическая разметка учитывает смысл выделенного фрагмента – цитата, сведения об авторе и т.п.

Современные стандарты языка HTML советуют использовать по-возможности логическую разметку, хотя физическое форматирование еще никто не отменял.



# теги физических стилей

<code>&lt;b&gt;...&lt;/b&gt;</code>	Полужирный шрифт	<code>&lt;big&gt;...&lt;/big&gt;</code>	Увеличенный шрифт
<code>&lt;i&gt;...&lt;/i&gt;</code>	Курсив	<code>&lt;small&gt;...&lt;/small&gt;</code>	Уменьшенный шрифт
<code>&lt;u&gt;...&lt;/u&gt;</code>	Подчеркивание	<code>&lt;sup&gt;...&lt;/sup&gt;</code>	Верхний индекс
<code>&lt;s&gt;...&lt;/s&gt;</code>	Зачеркнутый шрифт	<code>&lt;sub&gt;...&lt;/sub&gt;</code>	Нижний индекс
<code>&lt;tt&gt;...&lt;/tt&gt;</code>	Стиль печатной машинки	<code>&lt;blink&gt;...&lt;/blink&gt;</code>	Мерцающий текст (в Internet Explorer)

Можно комбинировать теги стилей:

`<b><i>Полужирный курсив</i></b>`

# Изменение шрифта

тег `<font> ... </font>`- изменяет размер, цвет и стиль текста.

Атрибуты:

`size` - размер шрифта 1-7. По-умолчанию `size=3`

Размер изменяется на 20%: 4 размер больше 3 на 20%,  
5 размер больше 4 на 20%

`size=4` – абсолютный размер,

`size= +1` – относительный (на 1 больше, чем базовый размер шрифта)

`color` - цвет текста,

`face` - название шрифта

`face="Comic Sans MS, Courier New"`

```
<html>
<head>
<title>Мой второй шаг </title>
</head>
```

```
<body>
```

Это обычный шрифт

```
<p>
```

```
<font size=5 color="#CC3399" face="Comic Sans MS, Courier
New">
```

Это измененный шрифт

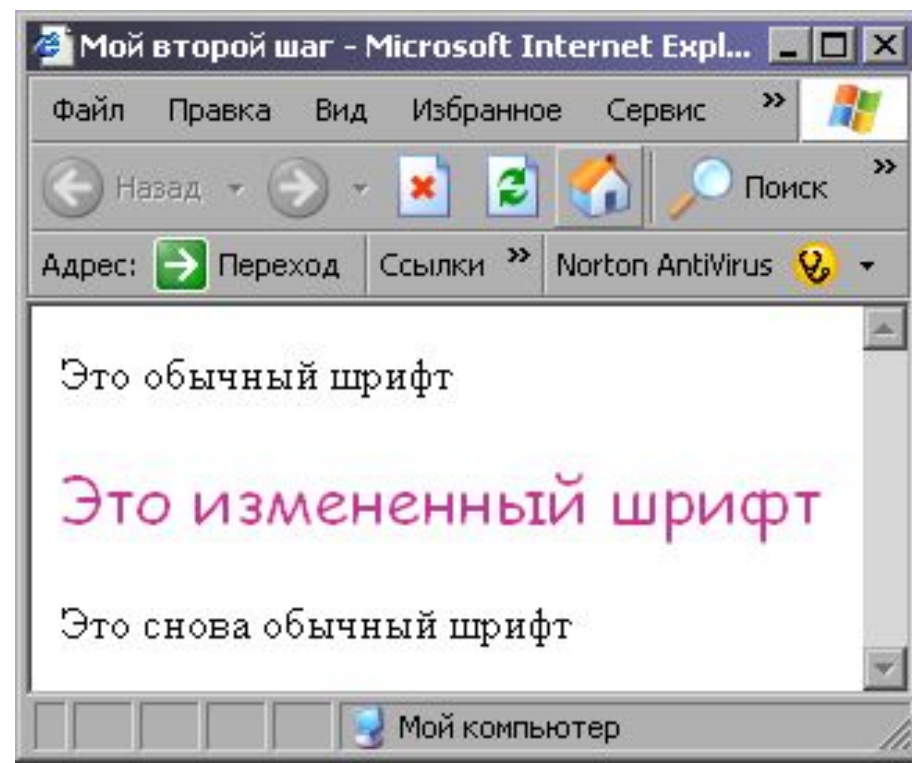
```
</font>
```

```
<p>
```

Это снова обычный шрифт

```
</body>
```

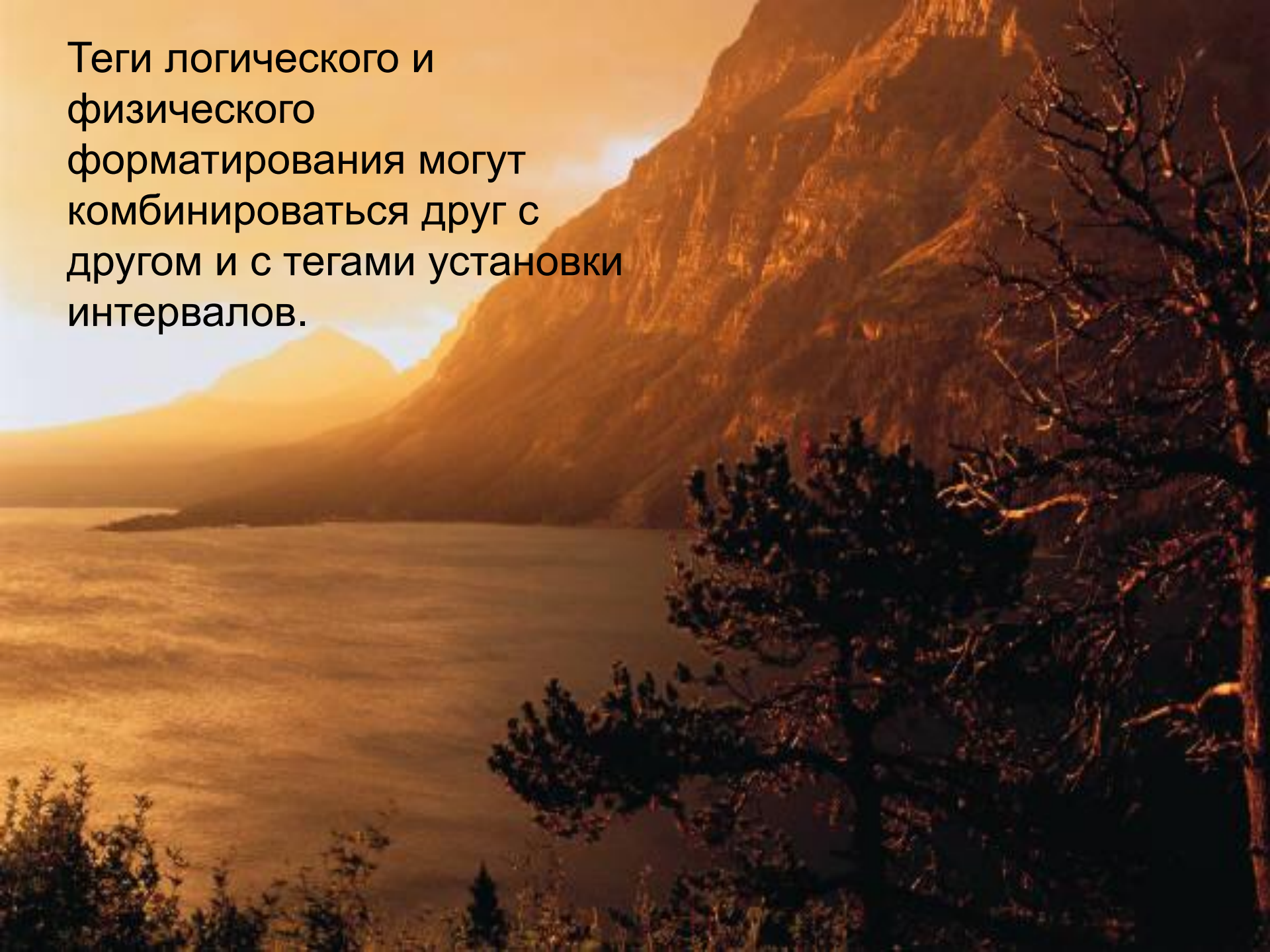
```
</html>
```



# ТЕГИ ЛОГИЧЕСКИХ СТИЛЕЙ

<code>&lt;abbr&gt;</code> <code>&lt;/abbr&gt;</code>	Аббревиатура	<code>&lt;acronym&gt;</code> <code>&lt;/acronym&gt;</code>	Сокращение, напр., HTML
<code>&lt;cite&gt;...</code> <code>&lt;/cite&gt;</code>	Библиографическая ссылка, исп. для записи названий книг и пр. (курсив)	<code>&lt;samp&gt;...</code> <code>&lt;/samp&gt;</code>	Текст с «буквальным» смыслом
<code>&lt;code&gt;</code> <code>&lt;/code&gt;</code>	Для текстов программ	<code>&lt;var&gt;...</code> <code>&lt;/var&gt;</code>	Для имен переменных
<code>&lt;em&gt;...</code> <code>&lt;/em&gt;</code>	Визуальное выделение (курсив)	<code>&lt;strong&gt;...</code> <code>&lt;/strong&gt;</code>	Логическое выделение (полужирный)
<code>&lt;dfn&gt;...</code> <code>&lt;/dfn&gt;</code>	Для специальных терминов	<code>&lt;kbd&gt;...</code> <code>&lt;/kbd&gt;</code>	Для технических терминов
<code>&lt;q&gt; ...</code> <code>&lt;/q&gt;</code>	Блок цитат (атрибут <code>cite="URL"</code> -источник цитаты)	<code>&lt;address&gt;...</code> <code>&lt;/address&gt;</code>	Адрес (курсив)

Теги логического и физического форматирования могут комбинироваться друг с другом и с тегами установки интервалов.



# Создание WEB-страниц в блокноте.

- В своей папке создать отдельную папку для файлов сайта.
- Открыть программу **БЛОКНОТ**.
- Написать в нем текст **WEB-страницы**.
- Сохранить этот текст в этой папке под любым именем с расширением **.html**.

- Перейти в эту папку.

Документ должен иметь значок



1.html

или



1.html

- Открыть этот документ. (Откроется программа-браузер).
- Если нужно внести изменения в документ, то выполнить команду **ВИД**⇒**Просмотр HTML-кода (Исходный текст)**.

Текст WEB-страницы откроется в блокноте.

- Внести в текст изменения. Сохранить его (**Ctrl+S**), перейти в программу-браузер и нажать кнопку

**ОБНОВИТЬ**.



# Итоги урока:

- Познакомились со структурой и основными командами форматирования WEB-документа.
- Узнали, как создавать простейшие WEB-документы.

# Задание на урок:

Создать две WEB-странички на произвольную тему

1. На первой применить физические стили форматирования (шрифт, курсив и т.п.), фон сделать цветным.
2. На второй применить логические стили форматирования. В качестве фона использовать картинку из файла.