

## «Основные теги»

[illegible]

Итак, ты уже знаешь, что любой HTML -документ— это обычный текстовый документ, просто размеченный особым образом. Соответственно, его можно создать в любом простейшем текстовом редакторе.

В операционной системе Windows простейшим текстовым редактором является программа **Блокнот**.

Запусти текстовый редактор (его всегда можно найти среди стандартных программ в главном меню операционной системы).

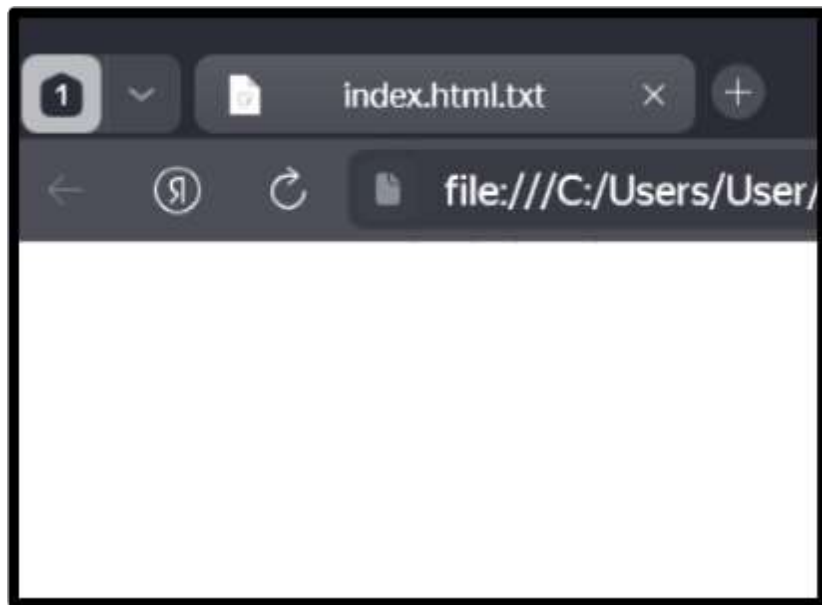
Выполни команду **Файл → Сохранить как...**

В открывшемся окне выбери место для сохранения файла и укажи имя файла — **index.html**.

Нажми кнопку **Сохранить**. Текстовый редактор пока не закрывай

Теперь давай разберёмся, что мы сейчас сделали. По сути, мы просто создали пустой текстовый файл и сохранили его с нужным расширением. Теперь этот файл можно открыть с помощью браузера.

- Открой свою папку, в которую ты сохранил файл.
- Найди файл `index.html` (он должен отображаться с пиктограммой браузера) и запусти его с помощью соответствующей программы, в нашем примере это Yandex браузер



Ты увидишь пустое окно браузера, и неудивительно, ведь мы ничего не написали в нашем файле! Но здесь важно другое — браузер теперь «умеет» отображать наш файл.

Если сейчас ты что-нибудь напишешь в текстовом редакторе и сохранишь сделанные изменения, а затем снова запустишь файл в браузере, увидишь, что браузер отобразит то, что ты написал. А знаешь, почему мы дали нашему файлу имя `index.html`?

Дело в том, что такое имя традиционно дают главному файлу сайта (файлу, с которого сайт должен запускаться).

Если ты уже поэкспериментировал с файлом, то заметил, что браузер отображает любое содержимое текстового файла, даже без всяких тегов. Ты же помнишь, что такое теги?

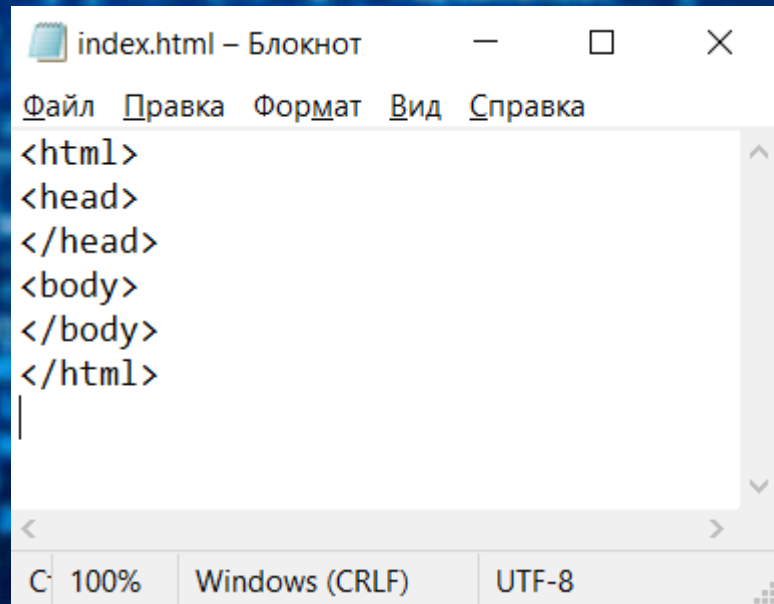
Зачем же нужно использовать теги?

- Во-первых, как бы ты ни форматировал текст в редакторе, браузер отобразит его по-своему — без форматирования, если только ты не укажешь ему правила отображения твоего текста специальными тегами.
- Во-вторых, для правильного отображения твоей страницы браузеру необходимо передавать некоторую служебную информацию, да так, чтобы он её понял, но эту информацию вовсе не нужно отображать посетителям сайта.



Поэтому каждый HTML-документ должен иметь правильную структуру, которую мы сейчас и зададим в нашем файле.

- Открой созданный файл в текстовом редакторе, если ты его уже закрыл.
- Наш файл должен быть пустым, поэтому, если ты что-то в него записывал, удали всю текстовую информацию.
- Запиши в файле теги начала и конца HTML-кода
- А между ними теги «головы» и «тела»



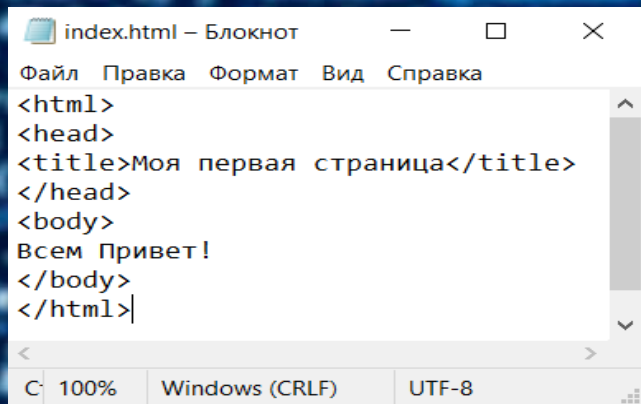
The screenshot shows a Notepad window with the title 'index.html - Блокнот'. The menu bar includes 'Файл', 'Правка', 'Формат', 'Вид', and 'Справка'. The text area contains the following HTML code:

```
<html>
<head>
</head>
<body>
</body>
</html>
|
```

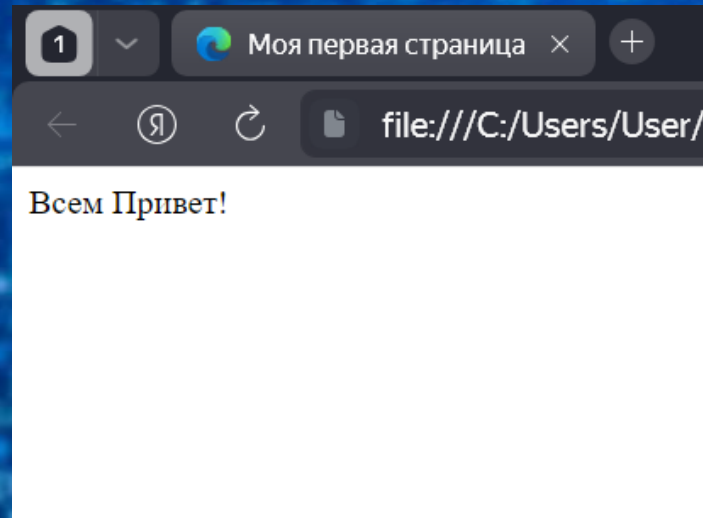
At the bottom of the window, there is a status bar with three sections: 'C' (indicating the current file), '100%' (zoom level), 'Windows (CRLF)' (line ending format), and 'UTF-8' (encoding).

Вот и всё! Структура документа задана, осталось  
наполнить её, а то пока что наша страничка  
при просмотре в браузере ничем не будет  
отличаться от предыдущего пустого  
документа.

- В разделе `<head></head>` помести  
теги `<title></title>` и запиши между ними  
название твоей страницы.
- В разделе `<body></body>` напиши фразу  
Всем привет!:



```
<html>
<head>
<title>Моя первая страница</title>
</head>
<body>
Всем Привет!
</body>
</html>|
```



Сохрани изменения в файле  
(**Файл** → **Сохранить** или сочетание клавиш  
**Ctrl+S**).

- Снова запусти файл в браузере, а если ты  
его не закрывал, то нажми кнопку **Обновить**  
или клавишу **F5**.

Вот так теперь выглядит наша страница:

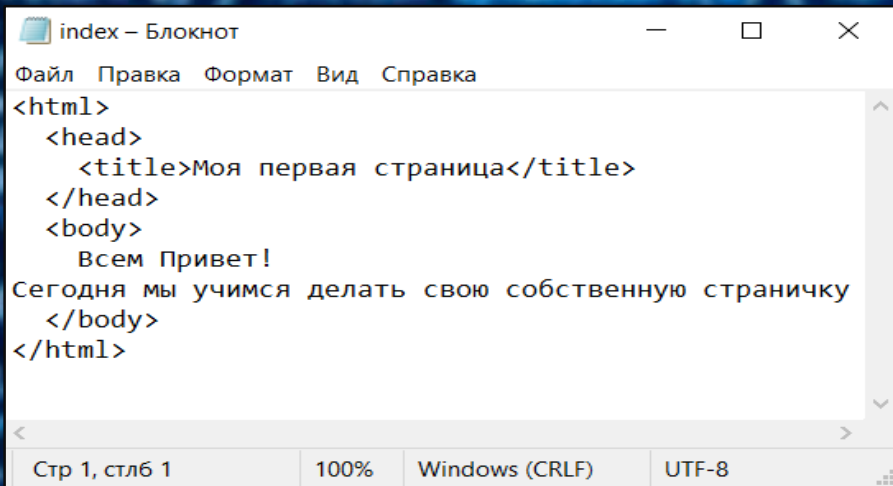
Как видишь, название страницы отобразилось  
на «корешке» вкладки, в которой она открыта.

А содержимое тегов `<body></body>` — в окне  
браузера.

Ну что ж, пора поработать над наполнением странички.

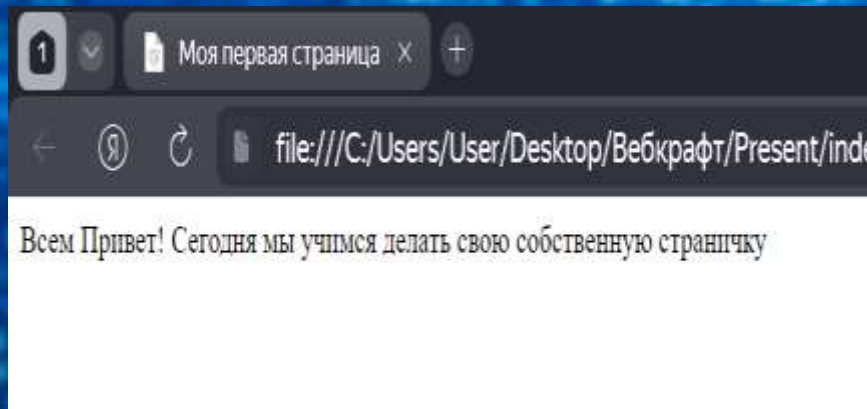
Давай разместим в теле страницы текст побольше.

- Открой файл `index.html` в текстовом редакторе, если уже закрыл его.
- Дополни текст, записанный в тегах **`<body></body>`**:
- Сохрани изменения в файле и посмотри на свою страницу в браузере:



```
index - Блокнот
Файл  Правка  Формат  Вид  Справка
<html>
  <head>
    <title>Моя первая страница</title>
  </head>
  <body>
    Всем Привет!
    Сегодня мы учимся делать свою собственную страничку
  </body>
</html>
```

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



Видишь, браузер отобразил всё содержимое сплошным текстом, хотя в текстовом файле были переносы строк, произошло это, потому что мы не разметили текст, то есть не указали браузеру, как его нужно отображать.



Основной тег для разметки текста — это абзац или параграф, он является парным и записывается так: `<p></p>`

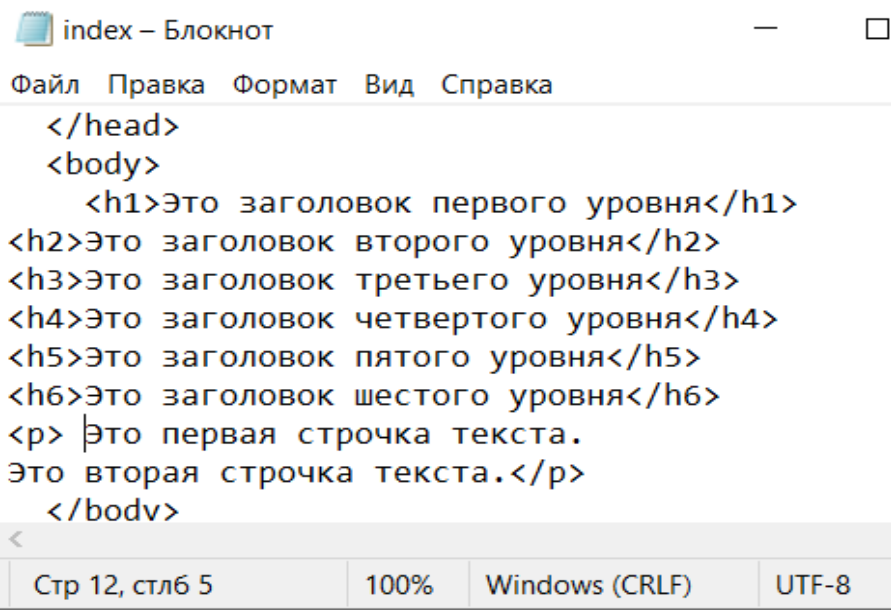
Каждый новый абзац текста должен быть заключён в эту пару тегов. Следует заметить, что параграфы

можно записывать и без закрывающего тега `</p>`, но лучше этого не делать, так как это не соответствует стандартам.

Для выделения заголовков предназначены теги от `<h1></h1>` до `<h6></h6>`.

Пара `<h1></h1>` определяет заголовок самого большого размера, а `<h6></h6>` — самого маленького.

Для наглядности рассмотрим пример:



```
index - Блокнот
Файл  Правка  Формат  Вид  Справка
</head>
<body>
  <h1>Это заголовок первого уровня</h1>
  <h2>Это заголовок второго уровня</h2>
  <h3>Это заголовок третьего уровня</h3>
  <h4>Это заголовок четвертого уровня</h4>
  <h5>Это заголовок пятого уровня</h5>
  <h6>Это заголовок шестого уровня</h6>
  <p> Это первая строка текста.
  Это вторая строка текста.</p>
</body>
```

Стр 12, стлб 5	100%	Windows (CRLF)	UTF-8
----------------	------	----------------	-------



Большинство структурированных текстов состоят из параграфов и заголовков. Упорядоченный контент делает чтение более лёгким и приятным. И теги разметки текста позволяют создавать иерархическую структуру.

**При этом следует соблюдать несколько правил при создании таких структур:**

- Предпочтительнее использовать `<h1>` только один раз на странице — это заголовок самого верхнего уровня, и все остальные заголовки располагаются ниже его в иерархии.
- **Нужно использовать заголовки в правильном порядке в иерархии. Не используй `<h3>` для создания подзаголовков при одновременном использовании `<h2>` для представления под-подзаголовков — это не имеет смысла и приведёт к странным результатам**

Из шести доступных уровней заголовка нужно стремиться использовать не более трёх на странице, если только нет острой необходимости в большем количестве уровней.

Теги заголовков также являются семантическими элементами, то есть придают тексту, который они обёртывают, роль (или значение) «заголовка определённого уровня на странице».

По умолчанию браузер придаст тексту в теге заголовка больший размер шрифта, чтобы он выглядел как заголовок. Но что ещё более важно, этот текст приобретёт семантическое значение.

Например, поисковые системы, индексирующие страницу, считают содержание заголовков важными ключевыми словами для влияния на ранжирование поиска страницы.

Также заголовки распознаются программами чтения с экрана.

Слабовидящие люди часто не читают веб-страницы — они слушают их вместо этого. Это делается с помощью программного обеспечения, называемого программой чтения с экрана. Среди различных используемых методов они формируют схему документа, считывая заголовки и позволяя своим пользователям быстро находить нужную им информацию. Если заголовки недоступны, пользователи будут вынуждены слушать весь документ.

При помощи тега можно задать любому фрагменту текста вид заголовка:

```
<span style=""font-size: 32px; margin: 21px 0;">Это заголовок верхнего уровня?</span>
```

Элемент `<span>` можно использовать для форматирования отрывков текста при помощи правил CSS, которые будут рассмотрены позже.

В данном примере для фрагмента текста установлен определённый размер шрифта и настроен отступ.

Здесь применили CSS, чтобы текст выглядел как заголовок верхнего уровня, но поскольку тег `<span>` не имеет семантического значения, этот текст не получит никаких дополнительных преимуществ, описанных выше



Зачастую текст необходимо как-то оформить: выделить важные моменты, установить подходящий шрифт или задать для определённого фрагмента специальное форматирование.

Для оформления фрагментов текста

существуют специальные теги, которые мы будем рассматривать далее.

Отдельные фрагменты текста можно выделить, придав им полужирное или курсивное начертание. Для этого предназначены теги `<b>`/`</b>` или `<strong>`/`</strong>` (полужирный текст) и `<i>`/`</i>` или `<em>`/`</em>` (курсив).

Почему же так получилось, что для одного и того же действия существует два различных тега? Теги `<b>` и `<i>` появились в эпоху, когда CSS поддерживался плохо или вообще не поддерживался, и предназначались, чтобы оформлять жирный текст и курсив. Изначально эти элементы являлись тегами физической разметки, то есть просто придавали тексту полужирное или курсивное начертание, без семантического акцента на его важность.

А теги `<strong>` и `<em>` изначально являются тегами логической разметки и обозначают, что этот текст выделен (`<em>`) или особо сильно выделен (`<strong>`).

Логическую разметку понимают поисковые роботы и речевые браузеры, поэтому в настоящее время рекомендовано использовать теги логической разметки.

Теги `<b>` и `<i>` НЕ читаются голосовыми браузерами, поэтому их лучше не использовать

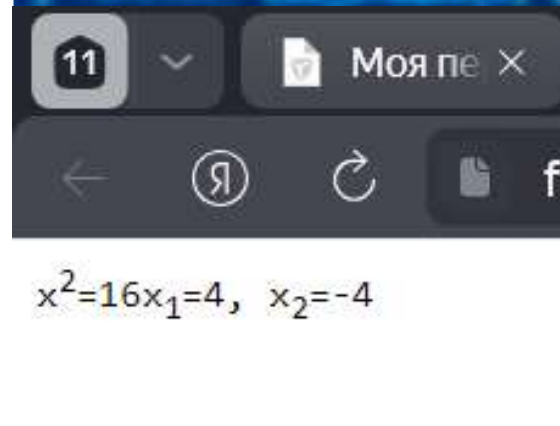
## Различия тегов `<b>` и `<strong>`, а также `<i>` и `<em>`

Существует ещё пара специальных тегов для оформления текста. Это `<sup>`/`</sup>` (верхний индекс) и `<sub>`/`</sub>` (нижний индекс). Чтобы понять, для чего они нужны, предположим, что тебе нужно разместить на веб-странице какие-то математические выражения, например, решение квадратного уравнения:

```
<pre>x2=16x1=4, x2=-4</pre>
```

Задача выполнена! Обрати внимание, в коде используется тег `<pre>`, и браузер отобразил его содержимое моноширинным шрифтом.

Подробнее об этом в следующем шаге

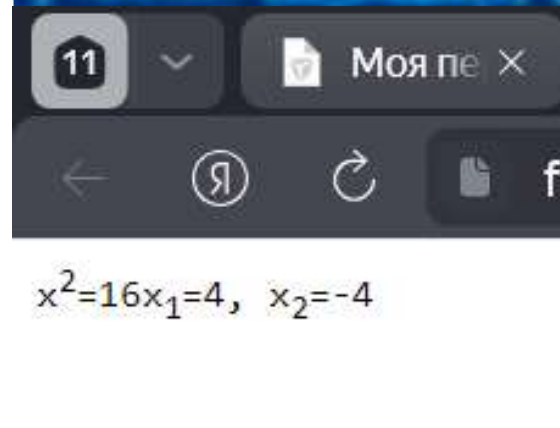


# Моноширные шрифты

Существуют теги, оформляющие текст моноширинным шрифтом, который имитирует стиль печатной машинки (все символы в таком шрифте имеют одинаковую ширину). Это теги логической разметки `<kbd></kbd>` и `<pre></pre>`.

Так как текст, выводится моноширинным шрифтом, эти теги обычно используют для оформления листингов — примеров какого-нибудь программного кода.

Они полностью сохраняют форматирование заключённого в них текста. То есть можно произвести форматирование прямо в текстовом редакторе — установить отступы, «лишние» пробелы и переносы строк, причём без использования специальных тегов `<br>` и последовательностей `&nbsp;`.





Для семантической разметки фрагментов компьютерного кода в тексте служат теги `<code></code>`. Они также отображают текст моноширинным шрифтом.

Тегами `<var></var>` выделяют в тексте имена переменных, текст отображается курсивом.

Для выделения результата, полученного в ходе выполнения программы, применяются теги `<samp></samp>`. Текст отображается моноширинным шрифтом.

HTML также имеет теги для оформления цитат и аббревиатур. Например, парный тег `<blockquote>` оформляет блочные цитаты внутри документа (абзац или несколько абзацев), выделяя их отступами и переносами строк.

Для этого тега доступен атрибут `cite`, в котором можно указать URL-адрес, указывающий на источник цитаты.

# Теги разметки

Тег	Вид тега	Действие/назначение
<p>	Парный	Создаёт новый абзац
<h1>.. <h6&gt;< td=""><td>Парный</td><td>Создаёт заголовок (от первого до шестого уровня — отличаются размером).</td></h6&gt;<>	Парный	Создаёт заголовок (от первого до шестого уровня — отличаются размером).
<b>	Парный	Задаёт полужирное начертание отрывка текста, не придавая акцент выделенному.
<strong>	Парный	Расставляет акценты в тексте, выделяя полужирным.
<i>	Парный	Выделяет отрывок текста курсивом, не придавая ему дополнительный акцент.
<em>	Парный	Выделяет фрагменты текста, курсивом.
<u>	Парный	Выделяет отрывок текста подчёркиванием.
<sup>	Парный	Задаёт надстрочное написание символов.
<sub>	Парный	Задаёт подстрочное написание символов.
<pre>	Парный	Выводит текст без форматирования, с сохранением пробелов и переносов текста. Может быть использован для отображения компьютерного кода.

# Теги разметки

Тег	Вид тега	Действие/назначение
 	Непарный	Принудительный переход на новую строку
<nobr>	Парный	Запрет перехода на новую строку
<hr>	Непарный	Создаёт горизонтальную черту-разделитель
<abbr>	Парный	Определяет текст как аббревиатуру. Поясняющий текст задаётся с помощью атрибута title.
<blockquote>	Парный	Оформляет блочные цитаты внутри документа (абзац или несколько абзацев), выделяя их отступами и переносами строк.
<q>	Парный	Используется для выделения коротких цитат внутри текста, текст внутри этого тега заключается в кавычки.
<cite>	Парный	Используется для указания источника цитирования. Содержимое отображается курсивом.



# Заключение

В следующем занятии разберем прикрепление медиафайлов на сайт  
До скорой встречи!