

HTML

Язык гипертекстовой разметки.

История HTML

- HTML 1.0 – 1989 г.
- HTML 2.0 – 1991 г.
- HTML 3.0 – 1995 г.
- HTML 4.0 – 1998 г. , HTML 4.01 –1999 г.
- HTML 5.0 – не утвержден по сей день

Браузеры должны знать какой стандарт HTML вы используете!!!

Переходная версия к HTML 4.01

Говорит браузеру, что здесь определяется тип документа для данной страницы

Это означает, что элемент `<html>` — корневой (первый) на вашей странице

Это просто означает, что стандарт HTML 4.01 является общедоступным

В этой части говорится, что мы используем версию HTML 4.01 и HTML разметка написана на английском языке

Вы можете напечатать все это в одну строку или, если хотите, разделить на несколько. Главное, убедитесь, что вы не разбиваете на несколько строк то, что взято в кавычки

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

Обратите внимание — это НЕ элемент HTML. После символа «!» сразу идет символ «<», что говорит вам, что это нечто иное

Это значение указывает на файл, который устанавливает особый стандарт

Чуть позже мы поговорим об этом более подробно

Браузеры должны знать какой стандарт HTML вы используете!!!

Строгая версия HTML 4.01

Эта часть такая же, как и для переходного DOCTYPE.

Слово Transitional исчезло.

Это строгая версия HTML 4.01 DOCTYPE.

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

В целом все выглядит очень похоже.

Помните, этот URL адрес определяет соответствие строгому HTML.

Браузеры должны знать какой стандарт HTML вы используете!!!

Для HTML 5.0 достаточно:

```
<!DOCTYPE html>
```

Он прощает все!!!

<http://validator.w3.org/>

проверка кода на валидность и соответствие стандартам

Основные понятия

Теги (от англ. «tag»-отметка)- операторы и команды HTML.

Все теги заключаются в

< >.

Следует заметить, что практически все теги надо закрывать:

<html>

Теги web-страницы

</html>

В закрывающем теге ставится /

<!-- Так добавляются комментарии-->

Основные понятия

Теги делятся на:

- **Блочные** элементы предназначены для **структурирования** основных частей вашей страницы, путём разделения содержимого на логически связанные блоки.
- **Строчные** элементы предназначены, чтобы разграничить часть текста и придать ему определённую функцию или смысл. Строчные элементы, как правило, содержат одно или несколько слов.

Структура документа

Любая web-страница задается следующими тэгами:

<html>

<head>

<title>Site name</title>

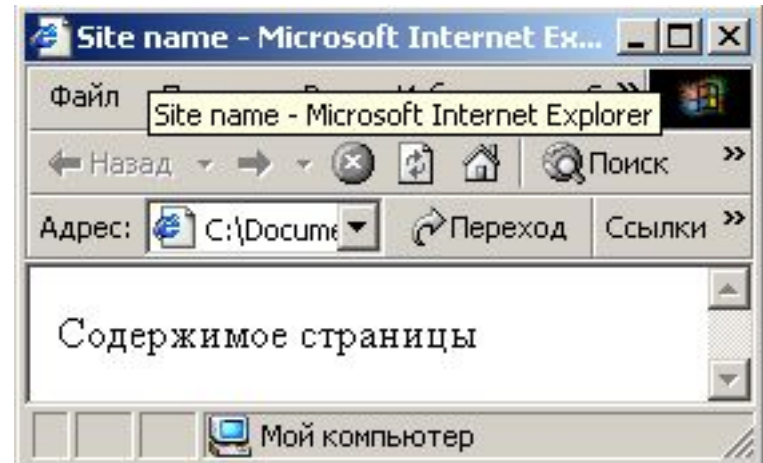
</head>

<body>

Содержимое страницы

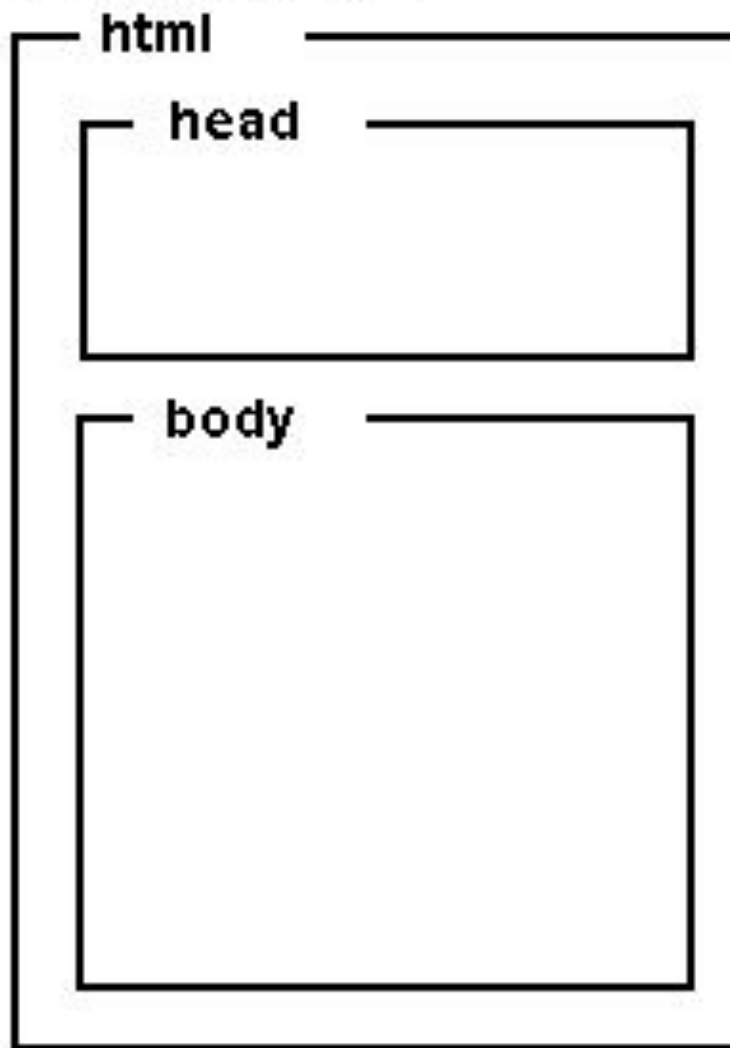
</body>

</html>



Структура документа

<!DOCTYPE HTML ...



Голова документа.

<head>

<title>Заголовок документа**</title>**

<meta http-equiv="Content-Type" content="text/html; charset=utf-8 или windows-1251">

<meta charset="utf-8"> - стандарт HTML 5

<meta name="keywords" content="тег, поисковая система">

<meta name="description" content="это сайт про теги">

<style> применяется для определения стилей элементов веб-страницы. **</style>**

</head>

Тело документа.

<body>

Предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера.

Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера

</body>

Тело документа

Атрибуты тега <body>

- **Alink** - Устанавливает цвет активной ссылки.
- **Background** - Задаёт фоновый рисунок на веб-странице.
- **Bgcolor** - Цвет фона веб-страницы.
- **Bgproperties** - Определяет, прокручивать фон совместно с текстом или нет.
- **Bottommargin** - Отступ от нижнего края окна браузера до контента.
- **Leftmargin** - Отступ по горизонтали от левого края окна браузера до контента.
- **Link** - Цвет ссылок на веб-странице.
- **Rightmargin** - Отступ от правого края окна браузера до контента.
- **Scroll** - Устанавливает, отображать полосы прокрутки или нет.
- **Text** - Цвет текста в документе.
- **Topmargin** - Отступ от верхнего края окна браузера до контента.
- **Vlink** - Цвет посещённых ссылок.

Теги форматирования текста

<h1>,...,**<h6>** - HTML предлагает шесть заголовков разного уровня, которые показывают относительную важность секции, расположенной после заголовка.

**** Устанавливает жирное начертание шрифта. ****

**** Предназначен для акцентирования текста. Браузеры отображают такой текст жирным начертанием. ****

**** Предназначен для акцентирования текста. Браузеры отображают такой текст курсивным начертанием. ****

<i> Устанавливает курсивное начертание шрифта. **</i>**

<q> Используется для выделения в тексте цитат. **</q>**

<blockquote> Предназначен для выделения длинных цитат внутри документа. **</blockquote>**

**
** устанавливает перевод строки в том месте, где этот тег находится.

<p align="left | center | right | justify"> Определяет текстовый абзац. **</p>**

<div align="center | left | right | justify" title="текст"> Является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого. Как правило, вид блока управляется с помощью стилей. **</div>**

Гиперссылки

Тег <a>:

- href=“путь” – связь с объектом, на который идет ссылка
- target=“название” – указание того окна, где будет открываться объект:
 - `_blank` - Загружает страницу в новое окно браузера.
 - `_self` -Загружает страницу в текущее окно.
 - `_parent` -Загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как `_self`.
 - `_top` -Отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как `_self`.

Пример:

```
<a href=“index.htm” target=“_blank”>Link</a>
```

```
<a href=“menu/menu.htm” target=“_blank”>Link</a>
```

```
<a href=“../menu/menu.htm” target=“_blank”>Link</a>
```

«Якоря»

Якорь – специальная метка в документе, невидимый объект, на который можно создать ссылку.

- Вставка якоря:

```
<a name=“название”></a>
```

- Ссылка на якорь:

```
<a href=“#название”>текст</a>
```

- Ссылка на якорь на другой странице:

```
<a href=“название страницы#название”>текст</a>
```

Вставка изображений

Тег :

- Src=“путь” – привязка к объекту
- Width=“число” – ширина
- Height =“число” – длина
- Border=“число” – толщина границ
- Bordercolor=“#цвет” – цвет границ
- Align=“значение”- расположение объекта:
 - Left
 - Right
- Alt= Альтернативный текст для изображения.
- Title = Название рисунка

Пример:

Тег закрывать **НЕ НАДО**

Списки

Нумерованный:

тег `` - ввод списка

тег `текст`- пункты списка

Пример:

```
<ol>
```

```
<li>text</li>
```

```
<li>text</li>
```

```
<li>text</li>
```

```
<li>text</li>
```

```
</ol>
```

Маркированный:

тег `` - ввод списка

тег `текст`- пункты списка

Пример:

```
<ul>
```

```
<li>text</li>
```

```
<li>text</li>
```

```
<li>text</li>
```

```
<li>text</li>
```

```
</ul>
```

Таблицы

Тег <table>:

- Width=“число” – ширина
- Height =“число” – длина
- Border=“число” – толщина границ
- Bordercolor=“#цвет”- цвет границ
- Bgcolor=“#цвет” – фон
- Cellspacing=“число”- расстояние между ячейками
- Align=“значение”- центровка

Строки и ячейки.

- Вставка строки – тег `<tr></tr>`
- Вставка ячейки – тег `<td></td>`:
 - `Width`=“число” – ширина
 - `Height` =“число” – длина
 - `Border`=“число” – толщина границ
 - `Bordercolor`=“#цвет” - цвет границ
 - `Bgcolor`=“#цвет” – фон
 - `Colspan`=“число”- кол-во столбцов, которое занимает ячейка.
 - `Rowspan` = устанавливает число ячеек, которые должны быть объединены по вертикали.

Пример

```
<table width=100% border=5 bordercolor=#000000 bgcolor="#FFF000"
  cellspacing=0>
```

```
<tr>
```

```
<td colspan=3>text</td>
```

```
</tr>
```

```
<tr>
```

```
<td width=30%>text</td>
```

```
<td width=40%>text</td>
```

```
<td width=30%>text</td>
```

```
</tr>
```

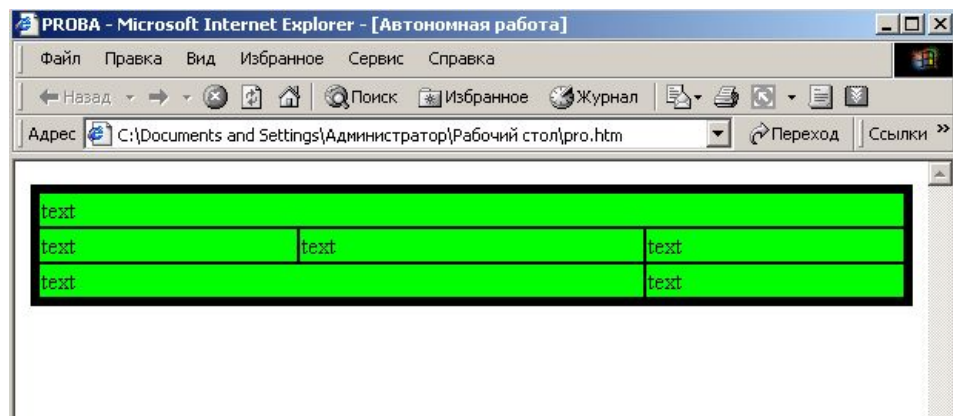
```
<tr>
```

```
<td colspan=2 width=70%>text</td>
```

```
<td>text</td>
```

```
</tr>
```

```
</table>
```



Задание

Наименование	Этапы	Часы	Стоимость
Дизайн	Главная страница		
	Внутренняя страница		
Верстка	Главная страница		
	Внутренняя страница		
Программирование	Подключение CMS		
	Дополнительные модули		
Сборка		Как пойдет	

Формы

Тег **<form>** устанавливает форму на веб-странице. Форма предназначена для обмена данными между пользователем и сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

- Для отправки формы на сервер используется кнопка Submit, того же можно добиться, если нажать клавишу Enter в пределах формы. Если кнопка Submit отсутствует в форме, клавиша Enter имитирует ее использование.
- Когда форма отправляется на сервер, управление данными передается программе, заданной атрибутом action тега **<form>**. Предварительно браузер подготавливает информацию в виде пары «имя=значение», где имя определяется атрибутом name тега **<input>**

Формы

Синтаксис - `<form action="URL"> ... </form>`

`<form action="handler.php">`

`<p>Как по вашему мнению расшифровывается аббревиатура "ОС"?`

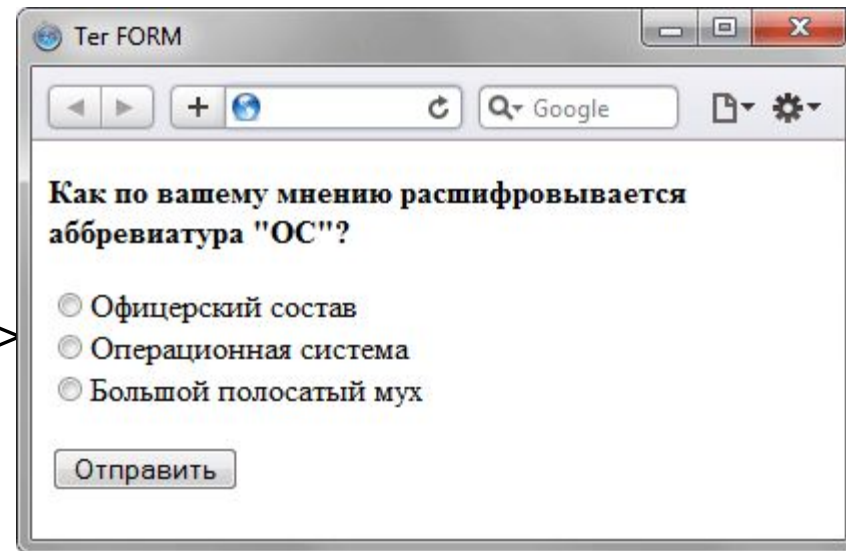
`<p> <input type="radio" name="answer" value="a1">Офицерский состав
`

`<input type="radio" name="answer" value="a2">Операционная система
`

`<input type="radio" name="answer" value="a3">Большой полосатый мух</p>`

`<p><input type="submit"></p>`

`</form>`



Формы

Тег **<input>** является одним из разносторонних элементов формы и позволяет создавать разные элементы интерфейса и обеспечить взаимодействие с пользователем. Главным образом **<input>** предназначен для создания текстовых полей, различных кнопок, переключателей и флажков. Хотя элемент **<input>** не требуется помещать внутрь контейнера **<form>**, определяющего форму, но если введенные пользователем данные должны быть отправлены на сервер, где их обрабатывает серверная программа, то указывать **<form>** обязательно. То же самое обстоит и в случае обработки данных с помощью клиентских приложений, например, скриптов на языке JavaScript.

Формы

Основной атрибут тега `<input>`, определяющий вид элемента — `type`.

Он позволяет задавать следующие элементы формы: текстовое поле (`text`), поле с паролем (`password`), переключатель (`radio`), флажок (`checkbox`), скрытое поле (`hidden`), кнопка (`button`), кнопка для отправки формы (`submit`), кнопка для очистки формы (`reset`), поле для отправки файла (`file`) и кнопка с изображением (`image`). Для каждого элемента существует свой список атрибутов, которые определяют его вид и характеристики.

```
<form name="test" method="post" action="input1.php"> <p><b>Ваше  
имя:</b><br> <input type="text" size="40"> </p> <p><b>Каким  
браузером в основном пользуетесь:</b><Br>  
<input type="radio" name="browser" value="ie"> Internet Explorer<Br>  
<input type="radio" name="browser" value="opera"> Opera<Br>  
<input type="radio" name="browser" value="firefox"> Firefox<Br> </p>  
<p>Комментарий<Br>  
<textarea name="comment" cols="40" rows="3"></textarea></p>  
<p><input type="submit" value="Отправить">  
<input type="reset" value="Очистить"></p> </form>
```

Формы

Ter INPUT

Navigation icons: back, forward, home, refresh, search, settings

Search: Google

Ваше имя:

Каким браузером в основном пользуетесь:

- Internet Explorer
- Opera
- Firefox

Комментарий

Стили внутри документа

<style> Применяется для определения стилей элементов веб-страницы. Тег необходимо использовать внутри контейнера **<head>**. Можно задавать более чем один тег **</style>**

<html>

<head>

<meta charset="utf-8">

<title>Тег STYLE</title>

<style type="text/css">

h1 { font-size: 120%; font-family: Verdana, Arial, Helvetica, sans-serif; color: #333366; }

</style>

</head>

<body>

<H1>Hello, world!</H1>

</body>

</html>

Это надо помнить!!!

Элемент `<html>`: не выходите из дома без него.

Всегда начинайте свою страницу с определения DOCTYPE, а сразу за ним указывайте `<html>`, который всегда должен быть корневым элементом вашей веб-страницы. Итак, после DOCTYPE тег `<html>` должен начинать вашу страницу, тег `</html>` - заканчивать, а все остальное должно быть вложено в него.

Всегда используйте `<head>` и `<body>`, чтобы HTML-код был лучше.

Только элементы `<head>` и `<body>` могут напрямую быть вложены в элемент `<html>`. Это означает, что все остальные элементы должны располагаться внутри элементов `<head>` или `<body>`.
Все без исключения!

Это надо помнить!!!

Что такое <head> без <title>?

В элементе <head> всегда помещайте элемент <title>. Это закон. Если вы этого не сделаете, то HTML будет признан не соответствующим стандартам. Элемент <head> - единственный, в который вы можете помещать элементы <title>, <meta> и <style>.

Наполняйте элемент <body> только полезными блочными элементами.

Непосредственно внутрь элемента <body> вы можете помещать только блочные элементы (<h1>, <h2> ... <h6>, <p>, <blockquote> и т. д.). Строчные элементы и текст необходимо сначала поместить в другой блочный элемент, и только затем они смогут войти в состав элемента <body>.

Это надо помнить!!!

Никогда не помещайте блочные элементы внутрь строчных.

Внутри строчных элементов вы можете помещать только другие строчные элементы или текст. Блочные элементы не могут входить внутрь строчных ни при каких обстоятельствах.

Не помещайте блочные элементы внутрь элемента `<p>`.

Абзацы предназначены для текста, поэтому не нужно помещать в них блочные элементы. Но, конечно же, любые строчные элементы можно использовать внутри абзацев (``, `<a>`, ``, ``, `<q>` и т. д.).

Это надо помнить!!!

Списки предназначены для перечисления пунктов.

Внутри элементов `` и `` разрешается использовать только элемент ``. Но зачем вам помещать в упорядоченный или неупорядоченный список что-то, кроме его пунктов?

Продолжим. Помещайте любые элементы внутрь элементов списка.

Внутри элементов списка `` вы можете помещать любой текст, а также строчные или блочные элементы .

Это надо помнить!!!

Кто бы мог подумать? Элемент `<blockquote>` любит только блочные элементы.

Элемент `<blockquote>` требует, чтобы внутри его было не менее одного блочного элемента. Пожалуйста, всегда помещайте ваш текст и другие строчные элементы в блочные элементы и уже эти блочные элементы добавляйте в `<blockquote>`.

Будьте внимательны, когда вкладываете строчные элементы друг в друга.

Несмотря на то что вы можете успешно вкладывать некоторые строчные элементы друг в друга, все же есть несколько случаев, в которых это действие нелогично и бессмысленно . Никогда не вкладывайте элемент `<a>` внутрь другого элемента `<a>`, потому что это может очень сильно запутать пользователей . И еще , не предусмотрено способа вкладывать строчные элементы в такие пустые элементы , как ``.

HTML 5!!!

HTML 5 — это не продолжатель языка разметки гипертекста, а новая открытая платформа, предназначенная для создания веб-приложений использующих аудио, видео, графику, анимацию и многое другое.

Ура, новые теги, облегчающие жизнь!!!

<article> <aside> <audio> <canvas> <command>
<datalist> <details> <figcaption> <figure> <footer>
<header> <hgroup> <keygen> <main> <mark>
<menu> <meter> <nav> <output> <progress> <rp>
<rt> <ruby> <section> <source> <summary>
<time> <video> <wbr>

HTML 5!!!

А это полезные теги, ориентированные на роботов, занимающихся интерпретацией текста, которые можно использовать вместо <div>

<article> задает содержание сайта вроде новости, статьи, записи блога, форума или др.

<aside> определяет блок сбоку от контента для размещения рубрик, ссылок на архив, меток и другой информации. Такой блок, как правило, называется «сайдбар» или «боковая панель».

<footer> задаёт «подвал» сайта или раздела, в нём может располагаться имя автора, дата документа, контактная и правовая информация.

<header> задает «шапку» сайта или раздела, в которой обычно располагается заголовок.

<nav> задает навигацию по сайту. Если на странице несколько блоков ссылок, то в **<nav>** обычно помещают приоритетные ссылки. Также допустимо использовать несколько тегов **<nav>** в документе.

HTML 5!!!

И еще немного полезной информации

[<<audio>](#) - Добавляет, воспроизводит и управляет настройками аудиозаписи на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег `<source>`. Внутри контейнера `<audio>` можно написать текст, который будет выводиться в браузерах, не работающих с этим тегом.

```
<audio src="URL"></audio>
```

```
<audio> <source src="URL"> </audio>
```

Атрибуты

autoplay - звук начинает играть сразу после загрузки страницы.

controls - добавляет панель управления к аудиофайлу.

loop - повторяет воспроизведение звука с начала после его завершения.

preload - используется для загрузки файла вместе с загрузкой веб-страницы.

src - указывает путь к воспроизводимому файлу.

HTML 5!!!

И еще немного полезной информации

[<video>](#) - Добавляет, воспроизводит и управляет настройками видеоролика на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег `<source>`.

```
< video src="URL"></ video >
```

```
< video > <source src="URL"> </ video >
```

Атрибуты

Autoplay - видео начинает воспроизводиться автоматически после загрузки страницы. **Controls** - добавляет панель управления к видеоролику. **Height** - задает высоту области для воспроизведения видеоролика. **Loop** - повторяет воспроизведение видео с начала после его завершения. **Poster** - указывает адрес картинки, которая будет отображаться, пока видео не доступно или не воспроизводится. **Preload** - используется для загрузки видео вместе с загрузкой веб-страницы. **Src** - указывает путь к воспроизводимому видеоролику. **Width** - задает ширину области для воспроизведения видеоролика.

HTML 5!!!

И еще немного полезной информации

[<<canvas>](#) - создает область, в которой при помощи JavaScript можно рисовать разные объекты, выводить изображения, трансформировать их и менять свойства. При помощи тега **<canvas>** можно создавать рисунки, анимацию, игры и др.

Атрибуты

- Height - Задает высоту холста. По умолчанию 300 пикселей.
- Width - Задает ширину холста. По умолчанию 150 пикселей.

СПАСИБО ЗА ВНИМАНИЕ!!!

Полезные ссылки

<http://htmlbook.ru/>

<https://webref.ru/>