

2. HTML: Понятие, стандарты, тэги и атрибуты

Понятие языка разметки гипертекста. Современные HTML-стандарты: HTML4 / 5, XHTML. Элементы HTML и их разновидности, структура тегов. Основные атрибуты тегов, классификация.

HTML: Понятие, стандарты, тэги и атрибуты

Понятие языка разметки HTML

HTML (HyperText Markup Language) – это не язык программирования (как, например, Python), а так называемый язык разметки. Он необходим для сообщения браузеру способа отображения посещаемой вами веб-страницы. Вместо того, чтобы показывать сплошной текст, **HTML** позволяет обернуть его в специальные элементы (**теги**). Это обеспечивает особое поведение передаваемой информации: она может выделяться курсивом, жирным текстом, транслироваться в виде таблиц, списков, рисунков и иных мультимедиа-данных. Теги способны нести как **структурный смысл** (например, показать, что перед нами таблица некоторой размерности с заголовком), так и **семантический** (выделить информацию определенным образом для поисковых роботов, чтобы те лучше индексировали ваш сайт, или браузеров).



Полезно знать HTML разработал британец Тим Бернес-Ли в конце 1980-х годов, а как официальный стандарт HTML функционирует с 1993 года. Им же опубликован первый в мире сайт – <http://info.cern.ch/hypertext/WWW/TheProject.html> - работающий по сей день. На нем содержится информация о технологии World Wide Web, базирующейся на протоколе HTTP, адресации URI, разметке HTML.

Сегодня Тим Бернес-Ли возглавляет W3-Консорциум (World Wide Web Consortium), основная задача которого: разработка и внедрение стандартов Интернета с учетом современных потребностей.

Изначально HTML был кроссплатформенным, что создавало определенные трудности. Разные браузеры отображали теги по-своему, некоторые элементы могли восприниматься не всеми программами. Причина: отсутствие единого стандарта, поддерживаемого разработчиками. Особую проблему доставлял Internet Explorer (Microsoft долгое время следовала своим наработкам в понимании HTML), что требовало от сайтостроителей учета разных видов браузеров, чтобы содержимое страниц выглядело более-менее одинаковым у пользователей. В последние годы данная проблема фактически решена, так как крупные игроки на рынке браузеров пришли к соглашению. На просторах Интернета еще можно встретить сайты, учитывающие старые браузеры, но это уже не обязательное требование к верстальщикам. Чаще всего вас просто попросят обновить свой браузер, чтобы вы смогли увидеть сайт в

задуманном виде (для эксперимента попробуйте зайти в любую социальную сеть с очень старого браузера).

Современные стандарты HTML

На протяжении курса мы будем придерживаться актуального стандарта **HTML5**. Он завершен в 2014 году и рекомендован для использования в современной веб-разработке. В него добавлены новые элементы, учитывающие специфику нынешнего сайтостроения и потребность в мультимедийных компонентах (аудио, видео, интерактивность).

До этого 17 лет главенствующим стандартом являлся **HTML 4** (с 1997 года). Он встречается и сегодня на ряде сайтов. Определить его можно на основании так называемого доктайпа.

Доктайп HTML4

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

Это первая строка любого **HTML-документа**, позволяющего браузерам понять способ работы с содержимым страницы. Хоть стандарт и является устаревшим, в нем содержится основной набор тегов, которыми пользуются при создании сайтов. Важно учесть, что часть элементов уже не рекомендована к применению и со временем будет убрана.

Несмотря на строгую типизацию в языке **HTML**, ряд браузеров достаточно лояльно относится к документам. Так, тег абзаца (обозначается `<p></p>`) разработчик вполне может написать по своему усмотрению:

- `<p>Абзац</p>` (написание в соответствии со стандартом);
- `<P>Абзац</P>` (браузеры редко учитывают регистр тегов);
- `<p>Абзац` (некоторые верстальщики могут забыть поставить закрывающий тег, что никак не смутит новую версию практически любого браузера).

Хорошая практика показывает, что нужно придерживаться некоего единого стиля. Чтобы как-то ограничить вольности программистов, внедрили стандарт **XHTML**. Он сочетает в себе **HTML** и **XML** форматы. Первоочередная задача данного стандарта – замена **HTML 4**. Однако после появления **HTML5** эта потребность несколько снизилась. Тем не менее, многие разработчики используют его и сегодня, чтобы гарантировать отсутствие ошибок в написании кода.

Ключевые особенности стандарта XHTML:

- все элементы пишутся строго в нижнем регистре;
- закрываются даже одиночные теги (например, ``);
- основная кодировка – **UTF-8**;
- все атрибуты записываются только в развернутом виде (например, если в **HTML5** отключить возможность выбора элемента из выпадающего списка можно указанием простого атрибута `disabled`, то в **XHTML** обязательно писать `disabled=«disabled»`).

Верстать код сайта на основании стандарта **XHTML** не обязательно, так как **HTML5** и современные браузеры не столь требовательны, но он может встретиться (и этого не нужно пугаться).

Охарактеризуем специфику HTML5:

- добавлены семантические элементы (**<nav>**, **<section>**, **<article>** и др.), которые облегчают чтение структуры страницы как разработчикам, так и поисковым машинам;
- внедрена поддержка векторной графики и специальных форматов (**svg**, **canvas**, **MathML**);
- представлены новые элементы управления (**dates**, **email**, **tel**);
- удалены устаревшие теги (**big**, **center**, **isindex**).



Итог

Таким образом, базис современной разработки – стандарт HTML5.

Консорциум Всемирной сети

Количество сайтов в сети настолько велико, что уже не хватает заложенного объема **IPv4-адресов** для их идентификации. С каждым годом число растет, что логично приводит к необходимости стандартизации их написания и отображения конечным пользователям.

Некоммерческая организация **W3-Консорциум** (www.w3.org) разрабатывает единые принципы и рекомендации для производителей Интернет-контента, чтобы обеспечить максимальную совместимость между продуктами, платформами, программами. Это позволяет делать Всемирную сеть максимально удобной и универсальной.



Помимо непосредственно стандартов **HTML** компания занимается утверждением и других

связанных положений:

1. CSS (каскадные таблицы стилей),
2. DOM (объектная модель документа),
3. PNG (формат хранения растровых изображений),
4. HTTP (протокол передачи данных),
5. RDF (модель представления метаданных),
6. XPath (язык для доступа к частям XML-документов) и др

Важно и то, что вы всегда можете проверить соответствие вашего кода стандарту при помощи **валидаторов**. В сети их огромное множество и для разных целей: **HTML, CSS, JavaScript**. Сервисы позволяют валидировать свой или чужой код. Для этого требуется ввести в поисковике требуемый (например, **HTML5** валидатор), вставить туда код и произвести проверку. Все нарушения будут подсвечены соответствующим цветом и подсказками. Ими пользуются все разработчики для верификации качества работы программистов. Все современные **IDE** дают возможность валидировать код (в большинстве случаев для этого потребуется установка плагина).

HTML теги. Структура тега



Тег – отдельный HTML элемент, позволяющий вкладывать или оборачивать части документа для определенного отображения или поведения.

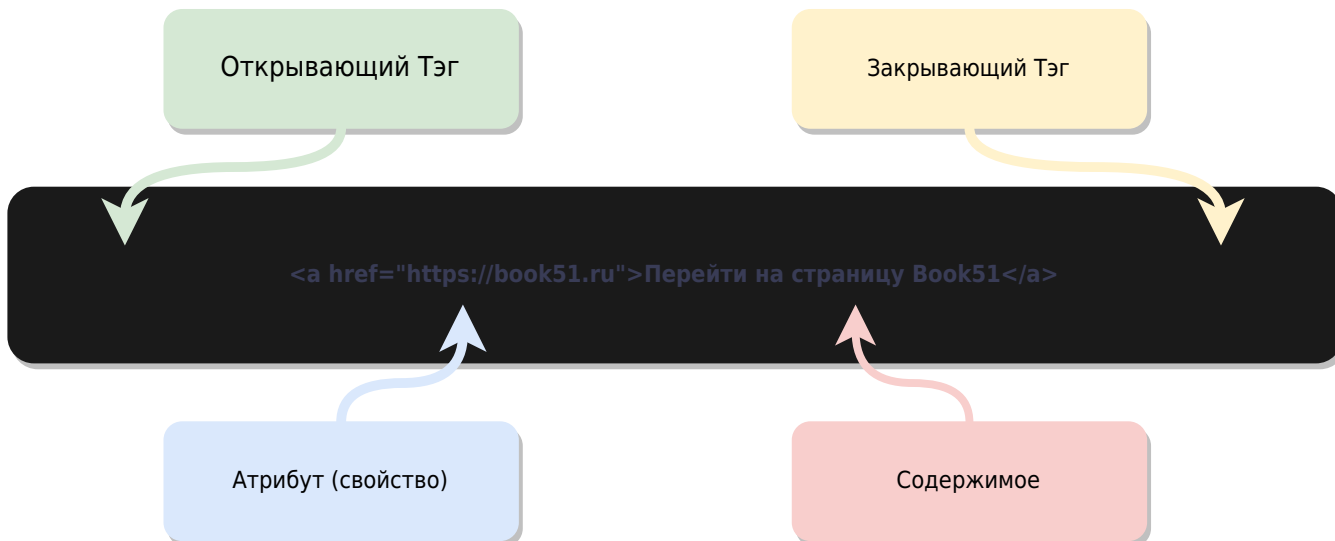
Говоря проще, при помощи тегов вы можете выделять текст, вставлять рисунки и видео, передавать служебную информацию, обозначать ссылки.

Теги могут дополняться атрибутами, свойствами. С их помощью можно модифицировать содержимое элемента, уникализировать его.

На текущий момент насчитывается около **120 тегов**, а если убрать устаревшие, то и того меньше. Их не так сложно запомнить, да и наиболее часто используемых несколько десятков.

Рассмотрим **типичную структуру web-элемента**:

1. **Открывающий тег** (все теги представляются в угловых скобках, в открывающем, если требуется, перечисляются атрибуты);
2. **Закрывающий тег** (присутствует не во всех тегах, идентифицируется косой чертой);
3. **Атрибут**, свойство (название свойства и его значение предпочтительно в двойных кавычках, возможно использовать и одинарные);
4. **Содержимое** (внутреннее содержимое тега, обычно в виде текста либо других вложенных тегов).



Подобным образом выглядит любой элемент web-страницы. При наличии нескольких атрибутов они разделяются пробелом. Теги не всегда могут иметь свойства или содержимое.

Первичное знакомство с разнообразием html-элементов можно осуществить на любом сайте. Если вы находитесь в браузере Chrome, то прямо сейчас можно нажать сочетание клавиш **CTRL + U** (откроются **Инструменты разработчика**) и просмотреть код данной страницы. Фактически, весь этот набор тегов и свойств в сочетании с **JavaScript** обрабатывается браузером, чтобы вы смогли удобно пользоваться текущим сайтом.

Сайт представляет собой совокупность таких страниц, связанных между собой внутренними и внешними гиперссылками.

Далее поговорим о **классификации тегов и атрибутов**. Более детальное их описание и примеры будут представлены в следующем уроке.

HTML теги. Парные и одиночные теги

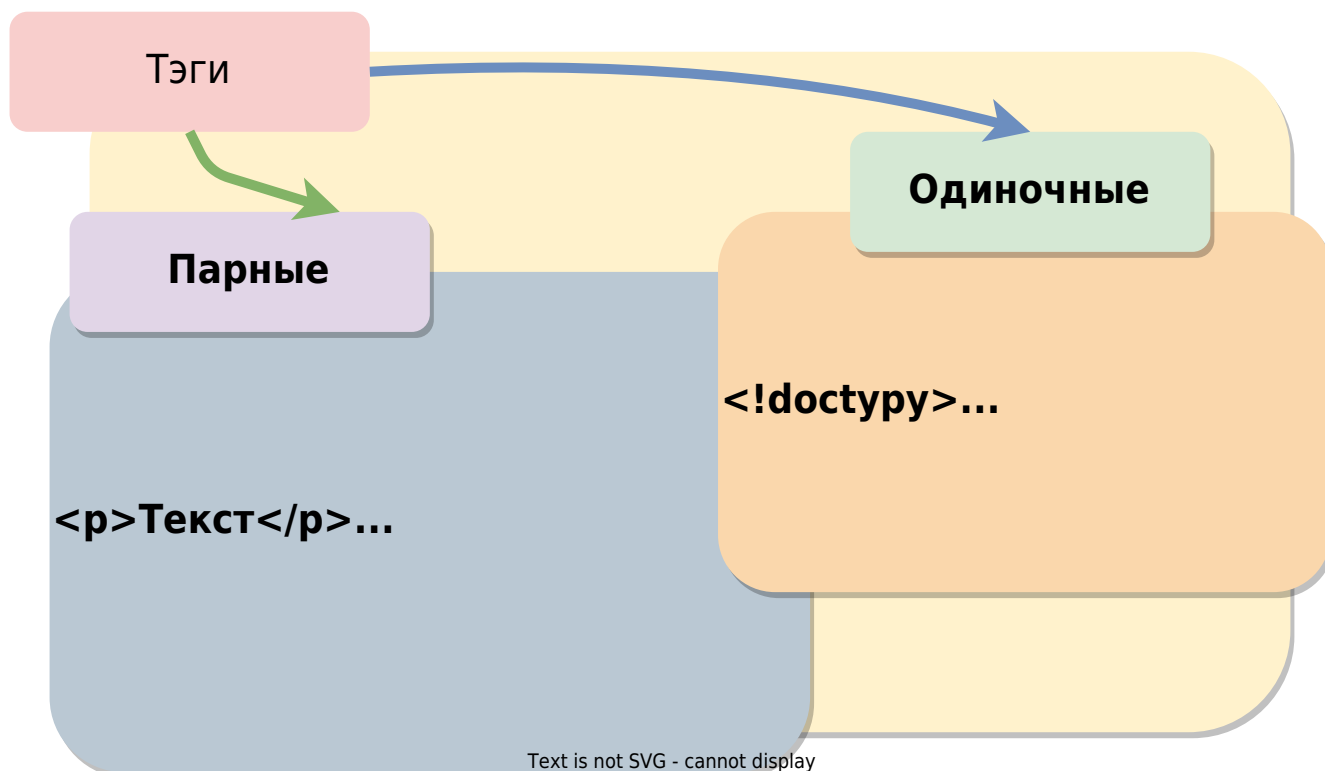
Как мы уже поняли выше, теги бывают парными или одиночными.

Одиночные теги

Одиночных элементов насчитывается **16 штук**. Наиболее часто используемые:

1. **<!doctype>** Некая инструкция браузерам для определения того, к какому типу относится документ, какую версию **HTML** использовать.
2. **<meta>** Необходим для задания дополнительной информации о странице, указывается в заголовке **HTML-документа**. Как и вышеупомянутый тег, не влияет на внешний вид страницы, так как носит служебный характер.
3. **<hr>** Задаёт горизонтальную линию для визуального отделения элементов страницы, применяется для отделения заголовка от основного текста, выделения логических блоков. По умолчанию представляет собой серую линию толщиной в 1 пиксель.
4. **
** Перенос нижеидущего содержимого на новую строку. Может потребоваться для избегания длинных полотён текста. Небольшие куски информации воспринимаются лучше, чем огромный абзац.

5. **** Чтобы вставить на страницу графический элемент. Рисунки можно демонстрировать в одном из следующих форматов: **jpg, png, gif**. Само изображение в документ не вставляется непосредственно, указывается ссылка на него (локальная или из другого источника), а уже браузер загружает картинку и демонстрирует пользователю.
6. **<input>** Применяется в формах и задает тип полей (могут быть текстовыми, числовыми, позволяют выбрать дату, файл, отмечать те или иные элементы списка и т.п.).



Парные тэги

Парные теги обязательно имеют **закрывающий тег** и содержимое (пусть даже и пустое). В **HTML** их больше всего (около **100**). В качестве примера приведем некоторые:

1. **<p>Текст</p>** Представляет заключенный внутри текст в виде отдельного абзаца. По умолчанию он имеет отступ и отделяется от предыдущего и последующего элементов дополнительным разрывом (это поведение можно поменять).
2. **Текст ссылки** Используется для задания и представления в документе ссылки (она может вести на другую страницу сайта, определенный участок на самой странице или любой адрес в Интернете).
3. **<small>Текст</small>** Делает текст меньшего масштаба по сравнению с основным.
4. **<mark>Текст</mark>** Выделяет текст по подобию маркера (аналог того, как на листах бумаги подчеркивают самое важное фломастером) желтым цветом.



С помощью CSS имеется возможность менять поведение практически всех тегов.

HTML теги. Блочные и строчные теги

Поведение любого тега сегодня (в стандарте **HTML5**) легко изменить по своему усмотрению. Деление элементов на блочные и строчные сложилось исторически (именно поэтому они изначально относятся к одному из этих типов). Сейчас таких типов больше, но их определение связано с каскадными таблицами стилей, о которых пойдет детальная речь в одном из последующих уроков.

Блочные теги занимают всю ширину страницы или родительского элемента. Если вы зададите два таких тега подряд, то каждый из них будет начинаться как бы с новой строки, абзаца.

Строчные теги не имеют строго размера. Он зависит от того, сколько символов в них содержится. Несколько подряд идущих строчных элементов не будут разделяться новыми строками, а будут выстраиваться друг за другом.

В **CSS-свойствах** отображение тегов можно менять по следующим типам:

1. **Блочные (display: block)** – получает свойство блочности, занимает всю ширину страницы;
2. **Строчные (display: inline)** – тег становится строчным;
3. **Флекс (display: flex)** – элемент приобретает свойство особого типа блоков – флекс (ведет себя как блок, но внутреннее содержимое гибко настраивается);
4. **Грид (display: grid)** – еще один блочный тип с особенностями внутреннего поведения элементов (строятся на основании «сетки», состоящей из строк и колонок);
5. **Таблица (display: table)** – тег и его содержимое будут наследовать свойства таблиц;
6. **Строчный блок (display: inline-block)** – блочный элемент ведет себя как строка, но сохраняет часть свойств (можно задать размер, границы).
7. **Спрятанный (display: none)** – делает элемент невидимым и полностью удаляет его отображение со страницы.



Существуют и другие свойства отображения, многие из которых узконаправлены и зачастую экспериментальны.

1. Блочный тэг

Самый популярный блочный тег времен HTML 4. Основная его задача - отобразить содержимое как блок. Он и сегодня входит в список часто используемых. Достаточно открыть код любой страницы в сети и проанализировать его. Почти везде

применяется максимально широко. Посмотрим на простой пример.

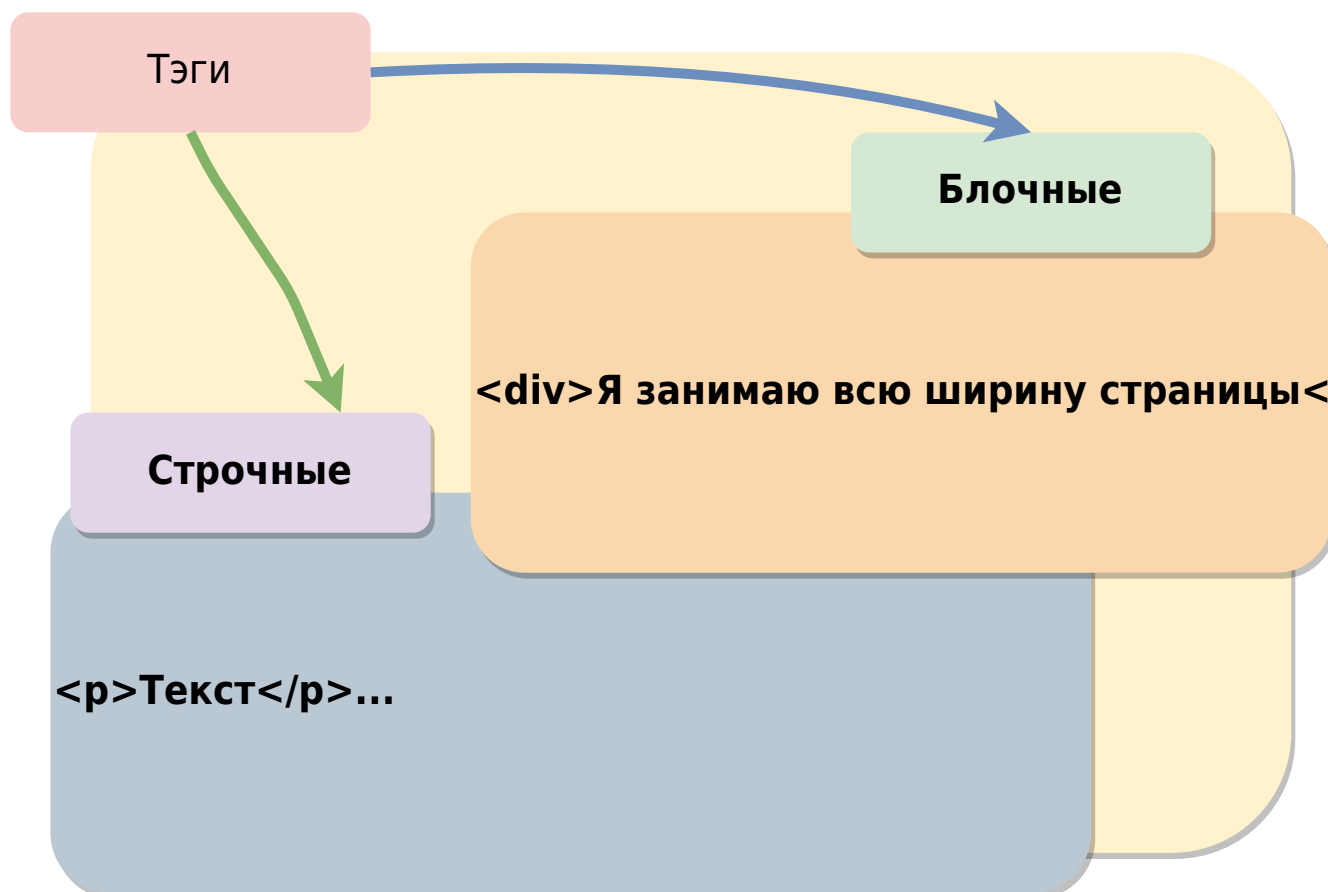
[Пример - HTML](#)

```
<div>Этот текст занимает всю ширину страницы</div>  
<div>Этот текст начнется с новой строки</div>
```

Если сохранить приведенный код в виде **html-файла** и открыть в браузере, то увидим, что куски текста начинаются с разных строк. Причина - в сути блочных элементов (они стремятся занять всю доступную ширину, которая им предоставлена).

Другие блочные теги:

1. **<table></table>** Позволяет рисовать таблицы
2. **<p>Параграф</p>** Обозначает параграф текста
3. **<h1>Большой заголовок</h1>** Для оформления заголовка самого верхнего уровня (в роли такого обычно выбирается основная статья страницы).
4. **<form></form>** Отражает на сайте форму (авторизации, регистрации, опроса и др.)



From: <http://git.wwooss.ru/> - worldwide open-source software

Permanent link: http://git.wwooss.ru/doku.php?id=software:development:ps_pycharm:html5:lesson:lesson.html_1&rev=1692542745

Last update: 2023/08/20 17:45

