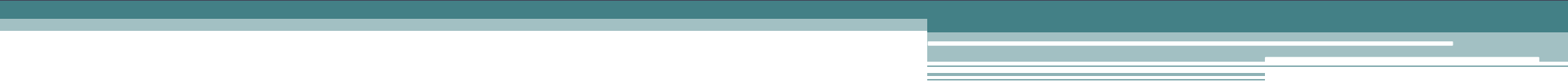


# Лекция 1. Введение в современный Web-дизайн. Структура HTML-документа. Тип документа. Метатэги



# Что такое WEB 2.0?

«Tim O'Reilly — What Is Web 2.0»  
от 30 сентября 2005 года

Тим О'Рейли дает такое определение «WEB 2.0 — методика проектирования систем, которые путём учета сетевых взаимодействий, становятся тем лучше, чем больше людей ими пользуются. Особенностью WEB 2.0. является принцип привлечения пользователей к наполнению и многократной выверке контента»

# Как выглядит WEB 2.0?

Скругленные углы, градиентные цвета, удобочитаемые шрифты, иконки, эффект отражения в картинках, разметка страницы в 1-2 колонки — всё это внешние общепринятые проявления WEB 2.0.

WEB 2.0 — это прежде всего удобство для пользователя. А значит, сайт в концепции WEB 2.0 должен сделать все возможное, чтобы пользователю было удобно, полезно и, конечно же, интересно находиться на таком сайте.

В последнее время графический облик классического дизайна веб 2.0 считается устаревшим и не креативным. Особенно это отражается в современной тенденции создания информативных сайтов, где главную роль играет простота, изящество, графичность и юзабилити.

# Принципы WEB 2.0

**Веб-сервисы.** Доступ к веб-сервисам осуществляется с помощью обычного интернет-браузера, пользователям не нужно устанавливать никаких дополнительных программ на своих компьютерах, не нужно заботиться о постоянных обновлениях. А выполнение каких-либо вычислений веб-сервисами не требует от компьютера пользователя никаких затрат ресурсов — все операции выполняет сервер.

**Mash-up** — возможность создать новый веб-сервис путем интегрирования программных возможностей нескольких других веб-сервисов.

# Принципы WEB 2.0

**Ajax** — методика применения технологий JavaScript и XML(JSON), которая позволяет загружать нужные данные в ответ на действия пользователя, не перезагружая веб-страницу целиком. Благодаря использованию этого подхода можно значительно ускорить работу пользователя с сайтом.

**RSS** — технология, основанная на XML, которая позволяет пользователям с помощью специальной программы-агрегатора просматривать новости с множества сайтов как единый поток — ленту новостей. Благодаря этой технологии пользователям больше не нужно просматривать десятки сайтов для того чтобы узнать последние новости.

# Принципы WEB 2.0

**Теги (ключевые метки)** позволяют более удобно идентифицировать и тематически сортировать контент (статьи, рисунки, мультимедиа файлы). В WEB 2.0 метки обычно представляются в виде так называемого «облака тэгов».

**Wiki-сайты** (наиболее ярким примером является Wikipedia) позволяют своим пользователям самим редактировать, добавлять или удалять информацию на сайте, создавать новые страницы. Таким образом пользователи более активно участвуют в наполнении сайтов нужной им же самим информацией.

# Принципы WEB 2.0

**Социализация.** Ведение личных сетевых дневников, блогов, — вот показательный пример социализации в Web 2.0. С помощью блога каждый пользователь может как-то выделиться из толпы, персонализировать свою определенную зону сайта — добавить личные аудио- и видеофайлы, изображения, публиковать свои статьи или делиться новостями. Кроме того, социализации способствует активное создание сообществ, в которых каждый пользователь может оставить свое сообщение, поделиться проблемой, получить множество разнообразных мнений и высказать свое по тому или иному поводу.

# Недостатки WEB 2.0

- зависимость сайтов от решений сторонних компаний, зависимость качества работы сервиса от качества работы многих других компаний;
- слабая приспособленность нынешней инфраструктуры к выполнению сложных вычислительных задач в браузере;
- уязвимость конфиденциальных данных, хранимых на сторонних серверах, для злоумышленников (известны случаи хищения личных данных пользователей, массовых взломов учётных записей блогов);
- серьёзнейшим недостатком сайтов web 2.0, где контент касается социально-значимых вопросов и допускаются пользователи под псевдонимами и анонимно;
- сбор статистики о пользователях, их предпочтениях и интересах, личной жизни, карьере, круге друзей могут помочь владельцу сайта манипулировать сообществом.



# История HTML

- 1991 – HTML
- 1994 – Консорциум W3 ([www.w3.org](http://www.w3.org))
- HTML 2.0
- Netscape Navigator
- 1995 – CSS
- 1996 – Internet Explorer
- 1997, январь – HTML 3.2
- 1997, декабрь – HTML 4.0
- 1999, декабрь – HTML 4.0.1
- 2002 – XHTML 1.0
- 2012 – HTML 5

# Консорциум W3C:

## <http://www.w3.org>

Разработка и продвижение открытых стандартов и рекомендаций для веб-технологий. Международная организация, основана в 1994

### **Более 400 участников:**

Adobe, AOL, Apple, Canon, CERN, Cisco, Dow Jones, Google, IBM, Intel, Microsoft, Mozilla, Nokia, Opera, Oracle, Samsung, Siemens, Sun, Yahoo...

[www.w3.org/Consortium/Member/List](http://www.w3.org/Consortium/Member/List)

### **Десятки технологий:**

HTML, XHTML, CSS, HTTP, URI, XML, DOM, MathML, PNG, SVG, XSLT...

[www.w3.org/Consortium/activities](http://www.w3.org/Consortium/activities)

Подробнее: [www.w3.org/Consortium](http://www.w3.org/Consortium)

# Тэг

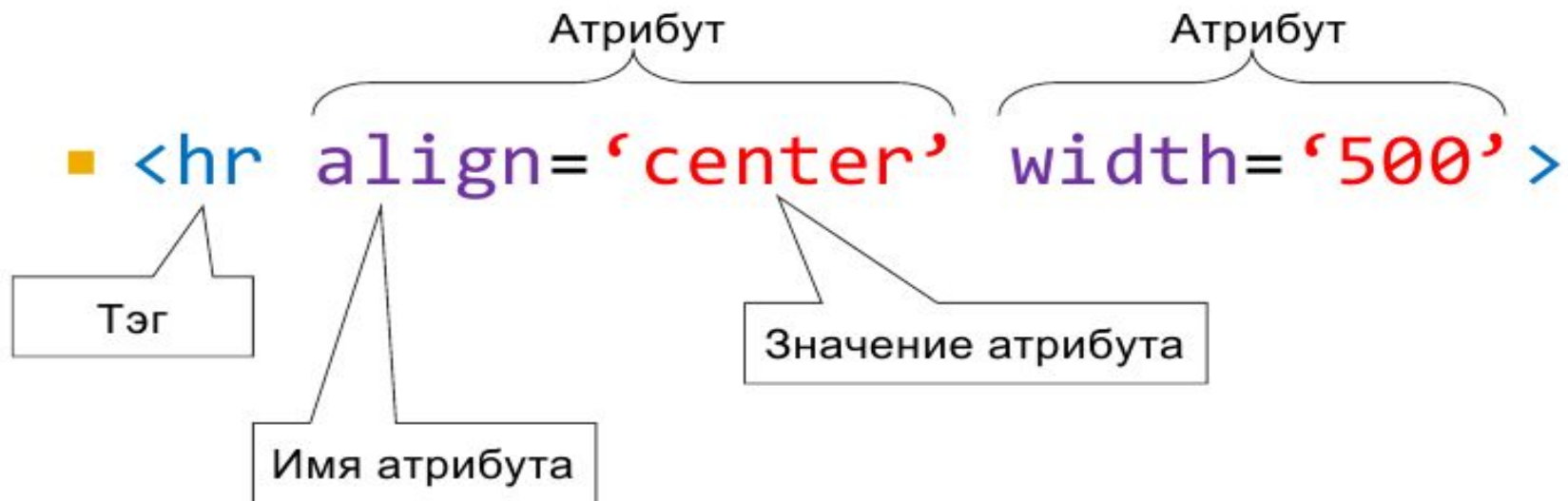
И Тэги бывают двух типов - одиночные и парные (контейнеры)



# Тэги. Атрибуты

## ТЭГОВ

Атрибуты можно подразделить на обязательные, они непременно должны присутствовать, и необязательные, их добавление зависит от цели применения тега



```

```

# Правила применения тэгов

- Атрибуты тэгов и кавычки
- Тэги можно писать как прописными, так и строчными символами
- Переносы строк
- Неизвестные теги и атрибуты
- Порядок тегов
- Закрывайте все теги

# Парные тэги

```
<div><p>Lorem <b>ipsum dolor</b> sit amet...</p></div>
```

```
<div><p>Lorem <b>ipsum <i>dolor</b> sit</i> amet...</div></p>
```

Рис. 1.1. Вложение тэгов, а - правильное, б - неверное

# Структура HTML-документа

- 1) строка объявления типа документа;
- 2) декларативный заголовок;
- 3) тело документа.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <title>Пример веб-страницы</title>
  </head>
  <body>
    <h1>Заголовок</h1>
    <!-- Комментарий -->
    <p>Первый абзац.</p>
    <p>Второй абзац.</p>
  </body>
</html>
```

# Объявление типа документа

```
<!DOCTYPE [Элемент верхнего  
уровня] [Публичность]  
" [Регистрация] // [Организация] // [  
Тип] [Имя] // [Язык] " " [URL] ">
```

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



# Объявление типа документа

**Элемент верхнего уровня** — указывает элемент верхнего уровня в документе, для HTML это тег <html>.

**Публичность** — объект является публичным (значение PUBLIC) или системным ресурсом (значение SYSTEM), например, таким как локальный файл. Для HTML/ХHTML указывается значение PUBLIC.

**Регистрация** — сообщает, что разработчик DTD зарегистрирован в международной организации по стандартизации (International Organization for Standardization, ISO). Принимает одно из двух значений: плюс (+) — разработчик зарегистрирован в ISO и - (минус) — разработчик не зарегистрирован. Для W3C значение ставится «-».

# Объявление типа документа

**Организация** — уникальное название организации, разработавшей DTD. Официально HTML/XHTML публикует W3C, это название и пишется в `<!DOCTYPE>`.

**Тип** — тип описываемого документа. Для HTML/XHTML значение указывается DTD.

**Имя** — уникальное имя документа для описания DTD.

**Язык** — язык, на котором написан текст для описания объекта. Содержит две буквы, пишется в верхнем регистре. Для документа HTML/XHTML указывается английский язык (EN).

**URL** — адрес документа с DTD.

# Допустимые DTD

DOCTYPE	Описание
<b>HTML 4.01</b>	
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"&gt;</code>	Строгий синтаксис HTML.
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"&gt;</code>	Переходный синтаксис HTML.
<code>&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd"&gt;</code>	В HTML-документе применяются фреймы.
<b>HTML 5</b>	
<code>&lt;!DOCTYPE html&gt;</code>	В этой версии HTML только один доктайп.

# Допустимые DTD

DOCTYPE	Описание
<b>XHTML 1.0</b>	
<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"&gt;</code>	Строгий синтаксис XHTML.
<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;</code>	Переходный синтаксис XHTML.
<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd"&gt;</code>	Документ написан на XHTML и содержит фреймы.
<b>XHTML 1.1</b>	
<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"&gt;</code>	Разработчики XHTML 1.1 предполагают, что он постепенно вытеснит HTML. Как видите, никакого деления на виды это определение не имеет, поскольку синтаксис один и подчиняется четким правилам.

# Правила XHTML

- 1) Все тэги и их атрибуты должны быть набраны в нижнем регистре (строчными символами).
- 2) Значения любых атрибутов необходимо заключать в кавычки.
- 3) Требуется закрывать все тэги, даже такие, которым не сопоставлен закрывающий тэг.
- 4) Должна соблюдаться правильная вложенность тэгов.
- 5) Нельзя использовать сокращенные атрибуты тэгов.
- 6) Вместо атрибута name следует указывать id (<a>, <frame>, <iframe>, <img> и <map>)
- 7) Следует определять DTD с помощью элемента <!DOCTYPE>
- 8) XHTML гораздо строже относится к ошибкам в коде; < и & везде, даже в URL, должны замещаться &lt; и &amp; соответственно

# Тэг HTML

Тэг `<html>` является контейнером, который заключает в себе все содержимое веб-страницы, включая тэги `<head>` и `<body>`.

## Атрибуты

<a href="#"><u>title</u></a>	Добавляет всплывающую подсказку на веб-страницу.
<a href="#"><u>manifest</u></a>	Указывает файл манифеста, необходимый для создания оффлайнового приложения.
<a href="#"><u>xmlns</u></a>	Указывает пространство имен для XHTML-документов.

`<html xmlns="http://www.w3.org/1999/xhtml">` - только для XHTML

`<html manifest="URL">` - только для HTML5

# Тэг HEAD

Тэг `<head>` предназначен для хранения других элементов, цель которых - помочь браузеру в работе с данными. Также внутри контейнера `<head>` находятся метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатэгам для получения описания сайта, ключевых слов и других данных.

Содержимое тэга `<head>` не отображается напрямую на веб-странице, за исключением тэга `<title>` устанавливающего заголовок окна веб-страницы.

Внутри контейнера `<head>` допускается размещать следующие элементы: `<base>`, `<basefont>`, `<bgsound>`, `<link>`, `<meta>`, `<script>`, `<style>`, `<title>`.

# Тег TITLE

Определяет заголовок документа. Элемент `<title>` не является частью документа и не показывается напрямую на веб-странице. В операционной системе Windows текст заголовка отображается в левом верхнем углу окна браузера. Допускается использовать только один тэг `<title>` на документ и размещать его в контейнере `<head>`.



# Тэг TITLE

- По тексту заголовка пользователь получает дополнительную информацию, что это за сайт, на котором он находится и как называется текущая страница.
- Большинство браузеров поддерживают возможность сохранения веб-страницы на локальный компьютер.
- При сохранении в разделе браузера «Избранное», адрес текущей страницы с ее заголовком помещается в список предпочитаемых ссылок.
- В результатах поиска по ключевым словам, поисковые системы используют заголовок страницы для указания ссылки на данный документ.

# Тэг META

`<meta>` определяет метатэги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатэгам для получения описания сайта, ключевых слов и других данных. Разрешается использовать более чем один метатэг, все они размещаются в контейнере `<head>`.

## Атрибуты

<a href="#">charset</a>	Задаёт кодировку документа.
<a href="#">content</a>	Устанавливает значение атрибута, заданного с помощью <code>name</code> или <code>http-equiv</code> .
<a href="#">http-equiv</a>	Предназначен для конвертирования метатега в заголовок HTTP.
<a href="#">name</a>	Имя метатега, также косвенно устанавливает его предназначение.

```
<META name="Author" content="Вася Пупкин">
```

```
<META http-equiv="Expires" content="Sun, 1 Nov 2009 16:20:47 GMT">
```

# Группа NAME

## 1) Метатэг Author и Copyright

Эти теги, как правило, не используются одновременно. Функция тегов — идентификация автора или принадлежности документа. Тег Author содержит имя автора Интернет-страницы, в том случае, если сайт принадлежит какой-либо организации, целесообразнее использовать тег Copyright. Выглядит данный тег следующим образом:

```
<meta name="author" content="Велимира Лисичкина" />
```

Кроме этого, теги Author и Copyright могут содержать дополнительный атрибут «lang», позволяющий определить язык, использующийся при указании значения свойства

```
<meta name="copyright" lang="ru" content="ПБОЮЛ Велимира Лисичкина" />
```

# Группа NAME

## 2) Метатэг Description

Данный тэг используется при создании краткого описания страницы, используется поисковыми системами для индексации, а также при создании аннотации в выдаче по запросу. При отсутствии тэга поисковые системы выдают в аннотации первую строку документа или отрывок, содержащий ключевые слова. Отображается после ссылки при поиске страниц в поисковике.

```
<meta name="description" content="Сайт об HTML и создании сайтов">
```

# Группа NAME

## 3) Метатэг Document-state

Метатэг Document-state также имеет значение при индексации страницы в поисковых системах. Тэг обладает двумя значениями — Static и Dynamic. Значение «Static» отмечает, что системе нет необходимости индексировать документ в дальнейшем, «Dynamic» позволяет регулярно индексировать Интернет-страницу.

```
<meta name="document-state" content="Dynamic" />
```

# Группа NAME

## 4) Метатэг Generator

Данный метатэг в первую очередь использовался разработчиками программ для редактирования веб-страниц с целью саморекламы — в качестве значения тэга разработчики таких программ, как правило, указывали название своего продукта

```
<meta name="generator" content="Macromedia Dreamviewer 4.0" />
```

# Группа NAME

## 5) Метатэг Keywords

Данный метатэг поисковые системы используют для того, чтобы определить релевантность ссылки. При формировании данного тега необходимо использовать только те слова, которые содержатся в самом документе. Использование тех слов, которых нет на странице, не рекомендуется. Рекомендованное количество слов в данном тэге - не более десяти. Кроме того, выявлено, что разбивка этого тэга на несколько строк влияет на оценку ссылки поисковыми машинами.

```
<meta name="Keywords" content="HTML, META, метатэг, тег, поисковая система">
```

# Группа NAME

## 6) Метатэг Resource-type

Тэг Resource-type описывает свойство или состояние страницы. Если значения тэга отличается от «Document», то поисковые системы его не индексируют.

```
<meta name="resource-type" content="document" />
```

## 7) Метатэг Revisit

Тэг позволяет управлять частотой индексации документа в поисковой системе. Для переиндексации сайта раз в две недели используется тэг следующего вида

```
<meta name="revisit" content="14" />
```



# Группа NAME

## 8) Метатэг url

Тэг прекращает индексацию страницы поисковой системой, и перенаправляет работа поисковой машины по указанной ссылке. Тэг применяется для отмены индексации «зеркала» и генерируемых страниц.

```
<meta name="url" content="http://ru.wikipedia.org/" />
```

## 9) Метатэг Subject

Определяет тематику документа. Практически бесполезен из-за отсутствия четкой и согласованной классификации тем в различных поисковых системах.

# Группа NAME

## 10) Метатэг Robots

Тэг формирует информацию о гипертекстовых документах, которая поступает к роботам поисковых систем. Значения тега могут быть следующими: Index (страница должна быть проиндексирована), Noindex (документ не индексируется), Follow (гиперссылки на странице отслеживаются), Nofollow (гиперссылки не прослеживаются), All (включает значения index и follow, включен по умолчанию), None (включает значения noindex и nofollow).

```
<meta name="robots" content="index,nofollow" />
```

# Группа HTTP-EQUIV

## 1) Метатэг Content-Language

Тэг позволяет указать язык, на котором создан документ

```
<meta http-equiv="content-language" content="ru" />
```

## 2) Метатэг Content-Script-Type

Определяет язык программирования сценариев. Если тег не прописан, следует указать язык программирования в каждом теге <script>. Тэг <script> обладает более высоким приоритетом по сравнению с Content-Script-Type.

# Группа HTTP-EQUIV

## 3) Метатэг Content-Style-Type

Указания языка таблицы стилей, по умолчанию значение - «text/css»

```
<meta http-equiv="Content-Style-Type" content="text/css">
```

## 4) Метатэг Content-Type

Определяет тип документа и его кодировку

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

# Группа HTTP-EQUIV

## 5) Метатег Expires

Метатеги Expires — управляет кэшированием. Если указанная в теге дата прошла, то браузер должен сделать повторный сетевой запрос, а не использовать копию из кэша. Если изначально указать прошедшую дату, то документ не будет кэшироваться. Некоторые поисковые роботы могут отказаться индексировать документ с устаревшей датой. Дата должна указываться в стандарте [RFC850].

```
<meta http-equiv="Expires" content="Wed, 26 Feb 1999 08:21:57 GMT">
```

# Группа HTTP-EQUIV

## 6) Метатег Pragma

Контроль кэширования при страницах, получаемых путем работы заданного скрипта.

```
<meta http-equiv="Pragma" content="no-cache">
```

## 7) Метатег Refresh

Задержка времени (в секундах) перед тем, как браузер обновит страницу. Кроме того, может использоваться автоматическая загрузка другой страницы

```
<meta http-equiv="refresh" content="5; url=http://www.example.com/" />
```

# Группа HTTP-EQUIV

## 8) Метатег Set-Cookie

Настройка cookie браузера

## 9) Метатег Window-target

Определение окна загружаемой страницы

## 10) Метатег Imagetoolbar

Отключает Панели управления изображениями. Обычно используют при выводе баннеров, фона картинки, карты изображения и др., когда вывод данной панели нежелателен

```
<meta http-equiv="imagemetoolbar" content="no" />
```