

Курс Web & DHTML

17 лекций + 17 практик

зачет

4 задания

- Верстка
- Фотогалерея
- Форум
- Модуль для Apache | WSDL пример

Разделы курса

- Верстка
- Клиентские скрипты
- Серверные скрипты

Используются знания, полученные при изучении курсов

- Протоколы Интернета
- Скрипты
- Языки сценариев
- Объектно-ориентированное программирование

План курса

1. Введение
2. Таблицы стилей, основные понятия
3. Таблицы стилей, рассмотрение базовых свойств
4. Верстка: практическое использование HTML и CSS
5. Настройки и дальнейшее развитие CSS
6. Обзор HTML 5

Клиентские скрипты

Серверные скрипты

План лекции № 1

Введение

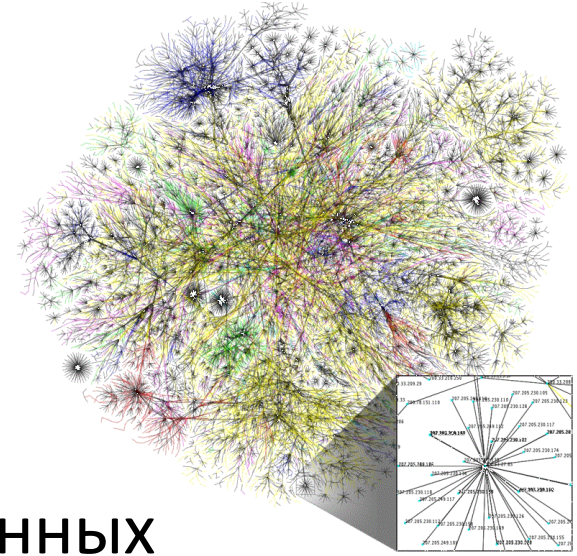
1. Воспоминание. Определение Интернета и перечисление его служб. Воспоминание о протоколе HTTP.
2. Языки разметки. Понятие разметки и языков разметки. История языков разметки.
3. Структура HTML документа. Декларация типа документа, элемент `<!DOCTYPE>`, режимы браузера. Элементы `html` и `head`.
4. Некоторые тэги и атрибуты. Блочные и строчные элементы.

www.opte.org/maps

www.opte.org/maps

Воспоминания

Что есть Интернет?



Интернет – всемирная система объединенных компьютерных сетей, построенная на базе стека протоколов TCP/IP и маршрутизации пакетов данных.

Интернет – это сеть сетей.

Службы Интернета – средства обеспечения определенных услуг для пользователей Интернета.

Воспоминания

Службы Интернета бывают разные

Службы Интернета – средства обеспечения определенных услуг для пользователей Интернета.

Интернет != Web

Интернет является инфраструктурой для служб Интернета.

- World Wide Web
- E-mail
- DNS
- и многие, многие другие...

Служба веб

Служба веб (англ. World Wide Web, всемирная паутина) – работающая по протоколу HTTP распределенная система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету.

The World Wide Web is a network of information resources. The Web relies on three mechanisms to make these resources readily available to the widest possible audience:

- A uniform naming scheme for locating resources on the Web.
- Protocols, for access to named resources over the Web.
- Hypertext, for easy navigation among resources.

The ties between the three mechanisms are apparent throughout this specification.

<http://www.w3.org/TR/html401/intro/intro.html>

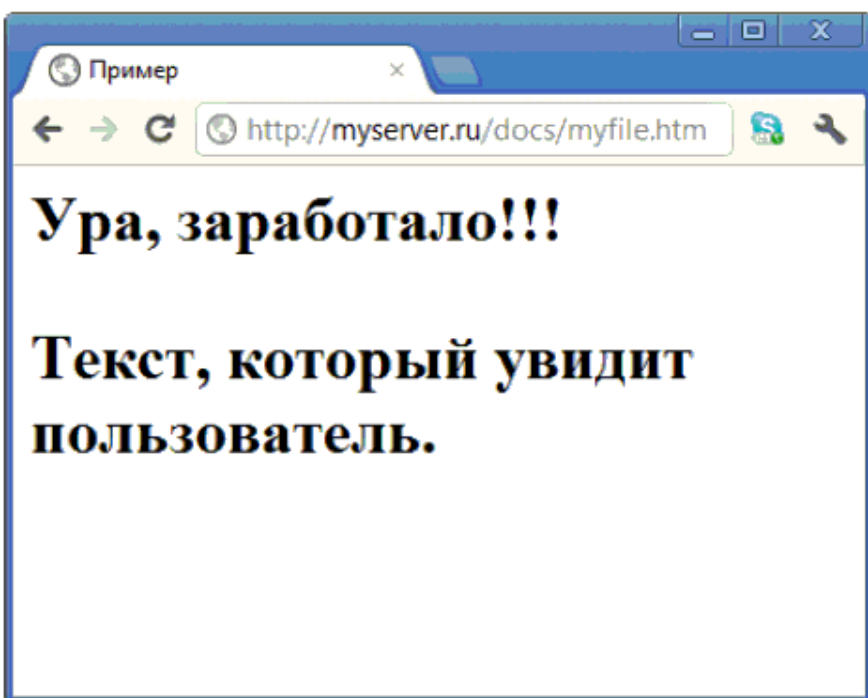


GET /docs/myfile.htm HTTP 1.1 →

← HTTP 200 OK



myfile.htm



```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    <h1> Ура, заработало!!!</h1>
    <p>Текст, который увидит
      пользователь.</p>
  </body>
</html>
```


План лекции

1. Воспоминание. Определение Интернета и перечисление его служб. Воспоминание о протоколе HTTP.
2. Языки разметки. Понятие разметки и языков разметки. История языков разметки.
3. Структура HTML документа. Декларация типа документа, элемент `<!DOCTYPE>`, режимы браузера. Элементы `html` и `head`.
4. Некоторые тэги и атрибуты. Блочные и строчные элементы.

HyperText Markup Language

HTML – язык, принятый в WWW для создания и публикации веб-страниц.

HTML предоставляет авторам средства для:

- форматирования текста,
- перехода к другим веб-страницам по гиперссылке,
- создания и заполнения форм,
- включения в веб-документы внешних объектов.

HyperText Markup Language

Фактически, современная Веб-страница формируется с помощью трех языковых средств:

- язык HTML используется для задания логической структуры документа,
- язык каскадных стилей CSS используется для задания способа отображения документа;
- языки программирования сценариев (чаще всего JavaScript) используются для написания сценариев.

Пример HTML-документа

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
  <head>
    <title>мой первый документ</title>
  </head>
  <body>
    <p>это html-документ.</p>
  </body>
</html>
```

История HTML

HTML был разработан Тимом Бернерсом-Ли (Tim Berners-Lee) 1986 – 1991 годах в CERN Laboratories.

HTML 2.0 — 22 сентября 1995 г.

HTML 3.2 — 14 января 1997 г.

HTML 4.0 — 18 декабря 1997 г.

HTML 4.01 — 24 декабря 1999 г.

XHTML 1.0 — 26 января 2000.

HTML 5 — 28 октября 2014 г.

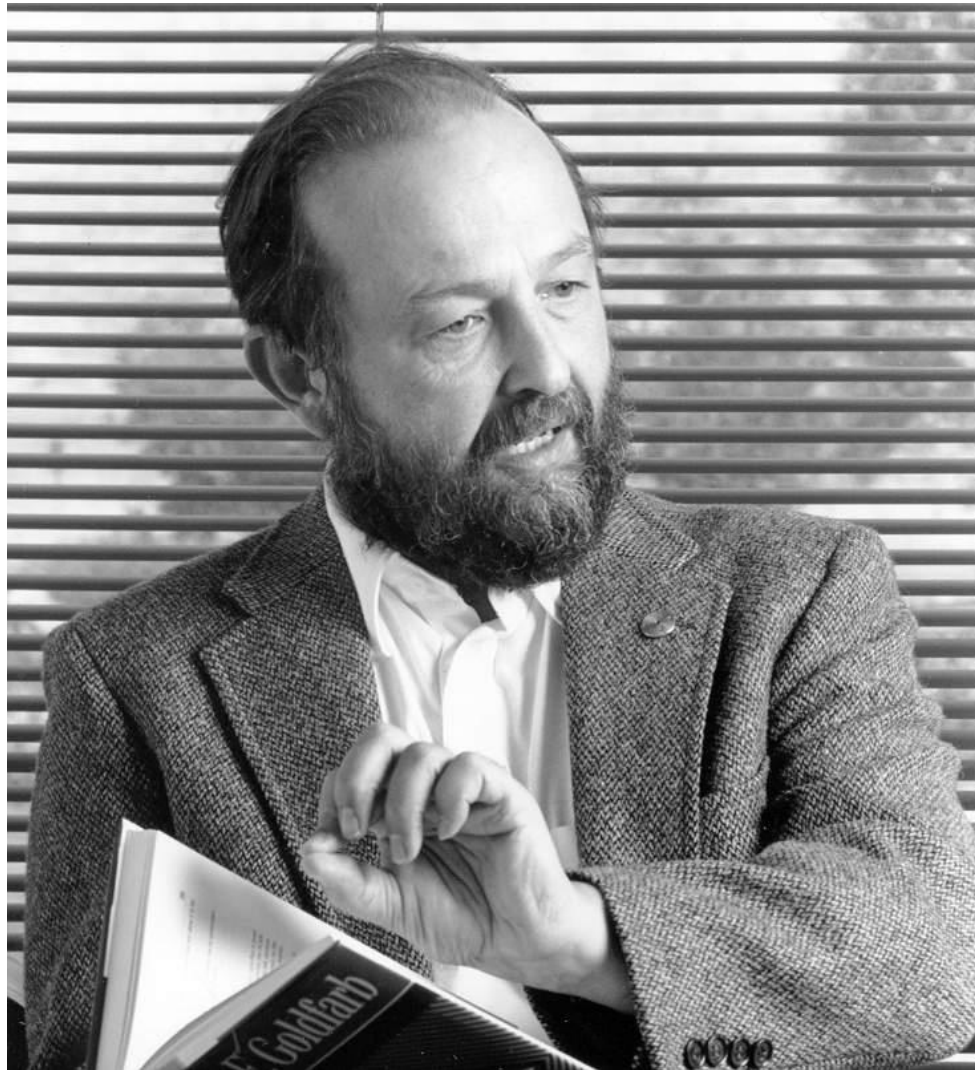
Языки разметки

Язык разметки – набор символов или последовательностей, вставляемых в текст для передачи информации о его выводе или строении.

Текстовый документ, написанный с использованием языка разметки, содержит не только сам текст (как последовательность слов и знаков препинания), но и дополнительную информацию о различных его участках – например, указание на заголовки, выделения, списки и т. д.

$$\int_{-\infty}^0 \frac{x^2+1}{\sqrt{x+2+y^2}} dx$$

Рождение языков разметки



Чарльз Голдфарб – отец языков разметки.

Хобби – составление
путеводителей

26. Left at light onto Jones Rd.

27. (Repeat instructions 20 - 26,
substituting "left" for "right".)

28. Second right.

1969 г. Голдфарб формулирует
основную концепцию будущего
языка GML.

1973 г. Голдфарб публично
представил GML.

Языки разметки: GML

GML (**Generalized Markup Language** – обобщённый язык разметки) – набор макросов, основной целью которых является реализация разметки, использующей теги для оформления текста на IBM.

:h1.Chapter 1: Introduction

:p.GML supported hierarchical containers, such as

:ol

:li.Ordered lists (like this one),

:li.Unordered lists, and

:li.Definition lists

:eol.

as well as simple structures.

Проект Aspen Movie Map, 1978

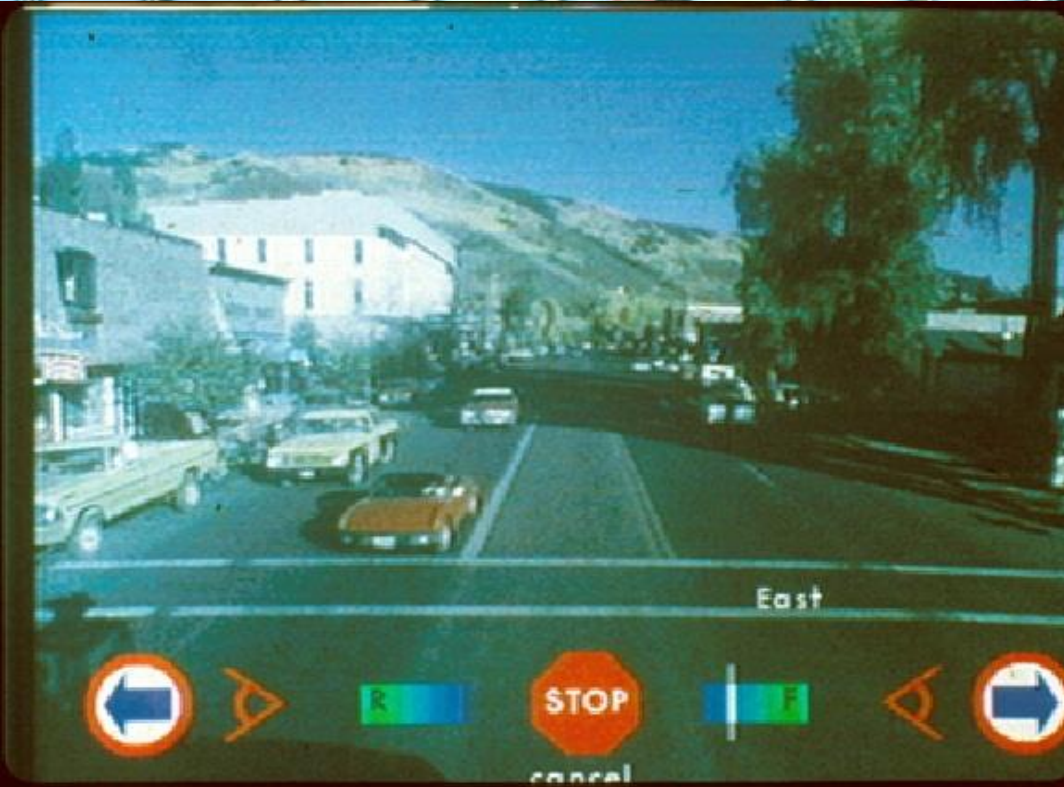


The image is a composite of three photographs illustrating the Aspen Movie Map project. The top-left photo shows a schematic street map of Aspen, Colorado, with 'MAIN STREET' highlighted in red. The top-right photo shows a person sitting in a chair, interacting with the system using a joystick and buttons, with a large screen displaying a 3D street view. The bottom photo shows a large screen displaying a 3D perspective view of a street in Aspen, with a control panel at the bottom featuring directional arrows, a 'STOP' button, and a 'cancel' button.

The Interactive Movie Map

- 1.0 Surrogate Travel
 - 1.1 Speed and Route Control
- 2.0 Augmentations
 - 2.1 Side Views
 - 2.2 Season Selection
- 3.0 Computer Generated Animation
 - 3.1 Ground Level Travel
 - 3.2 Aerial Travel
 - 3.3 Facade Billboarding

17



Языки разметки: SGML

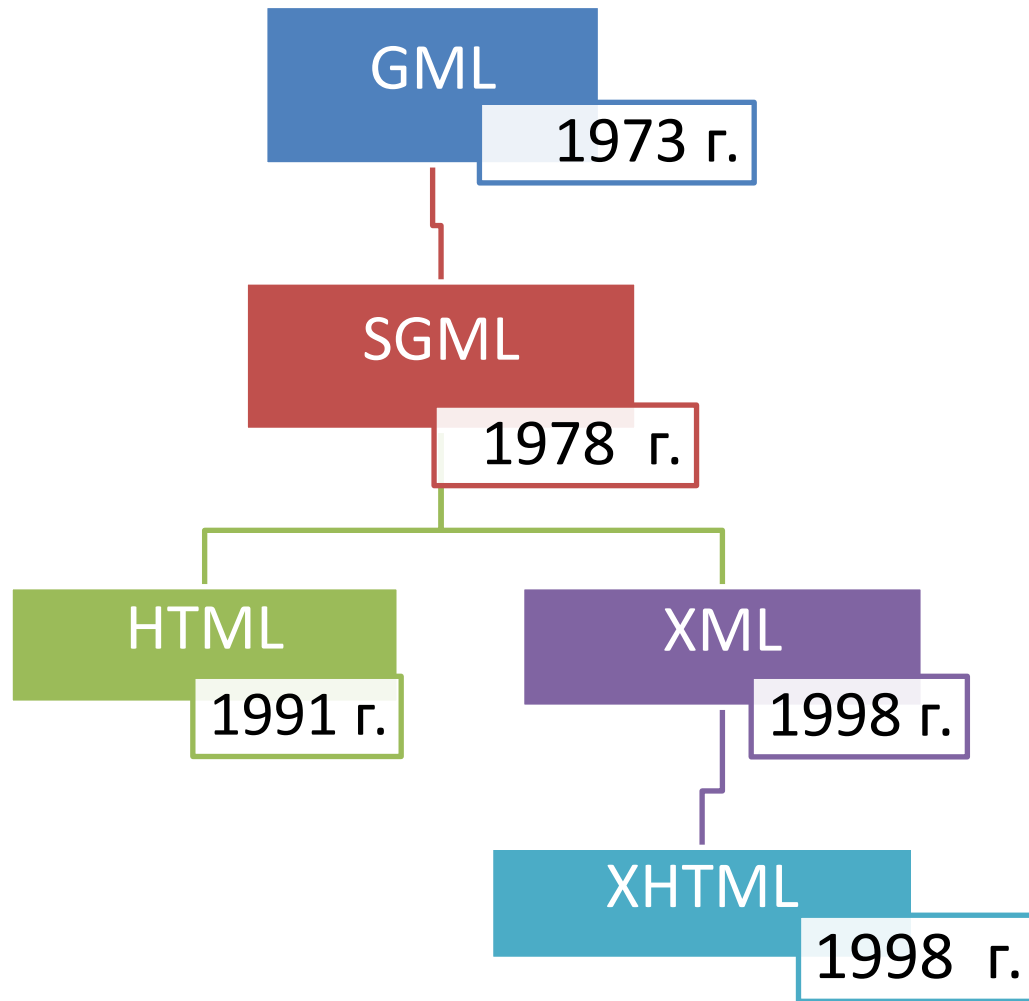
SGML (**Standard Generalized Markup Language** — стандартный обобщённый язык разметки) — метаязык, для определения языков разметки документов.

SGML предоставляет множество вариантов синтаксической разметки для использования различными приложениями.

```
<!DOCTYPE recipe SYSTEM "recipe.dtd">
<recipe type="dessert" ser="6" preptime="15">
<title>Two-Minute Fudge</title>

<ingredient-list>
<ingredient>1 pound of sugar</ingredient>
<ingredient>1/2 cup cocoa</ingredient>
...
</recipe>
```

Языки разметки: родословная



Языки разметки: XML

XML (eXtensible Markup Language) — упрощенный диалект языка SGML, предназначенный для описания иерархических структур данных в World Wide Web.

```
<?xml version="1.0" encoding="WINDOWS-1251"?>
```

```
<?xml-stylesheet href="my-style.css"?>
```

```
<knowledgeDatabase>
```

```
<tutorial>
```

```
  <title>Заметки об XSL</title>
```

```
  <author>Леонов Игорь Иванович</author>
```

```
</tutorial>
```

```
</knowledgeDatabase>
```


План лекции

1. Воспоминание. Определение Интернета и перечисление его служб. Воспоминание о протоколе HTTP.
2. Языки разметки. Понятие разметки и языков разметки. История языков разметки.
3. Структура HTML документа. Декларация типа документа, элемент `<!DOCTYPE>`, режимы браузера. Элементы `html` и `head`.
4. Некоторые тэги и атрибуты. Блочные и строчные элементы.

Составные элементы HTML-документа

Элементы и теги

****Элемент списка с конечным тегом****

****Элемент списка без конечного тега

Атрибуты

<img src="lion.gif" alt="Лев">

Специальные символы (сущности)

левая угловая скобка ("<") < амперсant ("&") &

символ авторского права ©©

Комментарии

Структура HTML-документа

- строка, содержащая декларацию типа документа
- заголовок документа (заключенный в теги <HEAD>...</HEAD>)
- тело документа (заключенного в теги <BODY>...</BODY> или <FRAMESET>...</FRAMESET>)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

```
<html>
```

```
<head>
```

```
<title>мой первый документ</title>
```

```
</head>
```

```
<body>
```

```
<p>Это текст абзаца. Здесь </ br> текст  
перенесен на следующую строку </p>
```

```
</body>
```

```
</html>
```

<!DOCTYPE>

Элемент <!DOCTYPE> предназначен для указания типа текущего документа — DTD (document type definition, описание 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">
```

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

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

Сравнение режимов браузера

Strict	Transitional
Чувствительность к регистру имен классов и идентификаторов	
Чувствительны к регистру	Нечувствительны к регистру
Пиксели по умолчанию	
Числа без указания единиц измерения игнорируются. width: 500 != width: 500px	Если в стилях в качестве единицы размера указано число без единицы измерения, считается, что значение задано в пикселах. width: 10 = width: 10px
Свойство display для изображений	
По умолчанию установлено как inline, при этом внизу картинок добавляется небольшой отступ.	По умолчанию установлено как block, при этом внизу картинок отступ отсутствует.

Стандартный и почти стандартный режим

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"  
"http://www.w3.org/TR/REC-html40/strict.dtd">
```

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

```
<table cellpadding="0"><tr><td>  
      
</td></tr></table>
```



Стандартный и почти стандартный режим

Стандартный: для изображений свойство `display = inline`.



Провекра Льва



Что делать?

```
TABLE IMG { display: block; } либо  
TD IMG { vertical-align: bottom; }
```

Зачем нужен Document Type Definition

`<u> Корректная вложенность тэгов </u>`

`<u> Ошибка </u> `

`<body>`

`<unknowntag>`

Сомнительный текст

`</unknowntag>`

`<table src="figure.jpg">`

`<td>`

`<tr>`Строка в ячейке`</tr>`

`</td>`

`</table>`

`</body>`

Document Type Definition

Язык схем DTD – язык, который используется для записи фактических синтаксических правил метаязыков разметки текста SGML и XML.

```
<!ELEMENT IMG - O EMPTY                                -- Embedded image -->
<!ATTLIST IMG
  %attrs;                                                -- %coreattrs, %i18n, %events --
  src            %URI;          #REQUIRED              -- URI of image to embed --
  alt            %Text;          #REQUIRED              -- short description --
  longdesc       %URI;          #IMPLIED               -- link to long description
                                                         (complements alt) --
  name           CDATA          #IMPLIED               -- name of image for scripting --
  height         %Length;       #IMPLIED               -- override height --
  width          %Length;       #IMPLIED               -- override width --
  usemap         %URI;          #IMPLIED               -- use client-side image map --
  ismap          (ismap)        #IMPLIED               -- use server-side image map --
>
```

Элемент HTML, атрибут manifest

```
<html manifest="demo.cache">
```

```
AddType text/cache-manifest .cache
```

```
CACHE MANIFEST
```

```
    /theme.css
```

```
    /main.js
```

```
NETWORK:
```

```
    login.asp
```

```
FALLBACK:
```

```
    *.php /offline.html
```

MIME тип text/cache-manifest

www.w3schools.com/html/tryhtml5_html_manifest.htm

Элемент HEAD

- единственный титул документа (TITLE);
- метаописатели документа (META);
- базовый URI внешних ссылок (BASE);
- ссылки на другие документы (LINK);
- внутренние таблицы стилей (STYLE);
- сценарии клиента (SCRIPT).

Метаописатели документа

http-equiv заголовок сообщения HTTP

name название свойства

content значение свойства

```
<meta name="Description"  
content="Web developer handbook">
```

```
<meta name="Keywords"  
content="Web, HTML, CSS, XML">
```

```
<meta name="Robots"  
content="index, nofollow">
```


Метаописатели документа

http-equiv заголовок сообщения HTTP

name название свойства

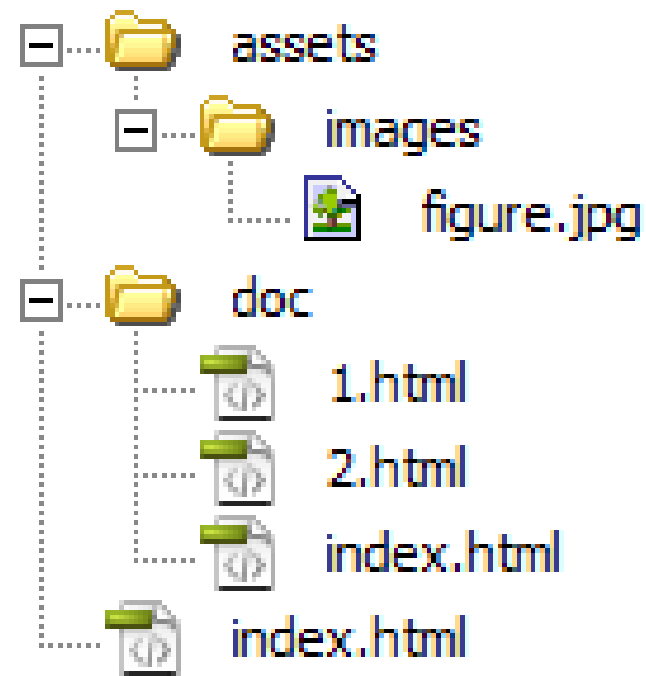
content значение свойства

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

```
<meta http-equiv="expires"  
  content="Sun, 01 Jan 2013 07:01:00 GMT">
```

```
<meta http-equiv="Cache-Control"  
  content="max-age=time, proxy-revalidate">
```

Элемент BASE



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<base href="/assets/images/">
```

```
</head>
```

```
<body>
```

```
<p></p>
```

```
</body>
```

```
</html>
```

```
<head>
```

```
<base href="doc/index.html">
```

```
</head>
```

```
<body>
```

```
<p>
```

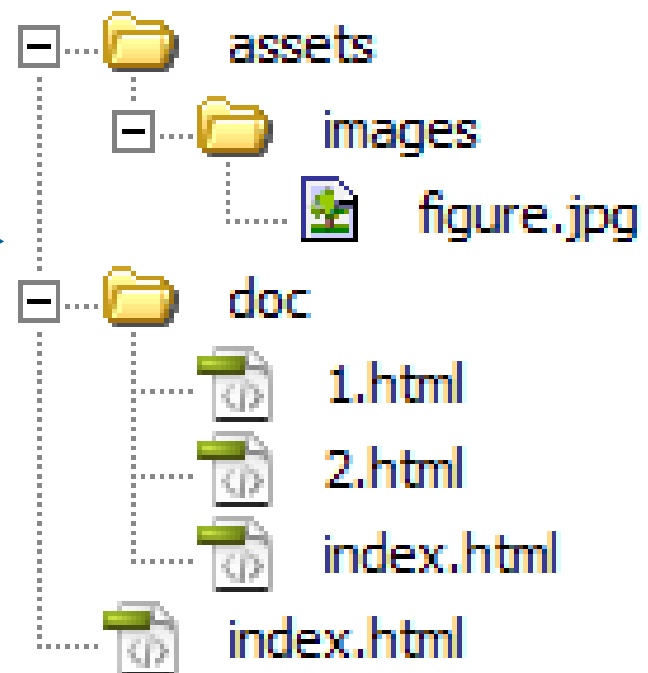
```

```

```
<a href="../../../2.html">Соседний документ</a>
```

```
<a href="../../../index.html">Вернуться на
главную страницу</a></p>
```

```
</body>
```



<head>

<base href="/doc/index.html">

</head>

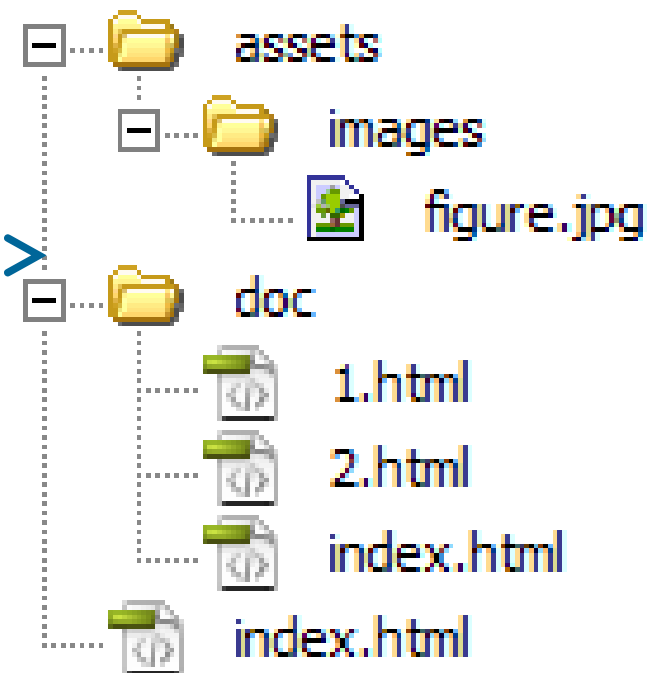
<body>

<p>

Соседний документ

Вернуться на главную страницу</p>

</body>



Элемент LINK

rel – прямая ссылка, отношения между текущим документом и файлом, на который делается ссылка

rev – обратная ссылка

href – путь к связываемому файлу

```
<link rel="stylesheet" href="main.css" type="text/css">
```

```
<link rel="glossary" href="glossary.html">
```

```
<link rev="subsection" href="section2.html">
```

```
<link rel="shortcut icon" href="img/ic.ico">
```

Элемент SCRIPT

async	Загружает скрипт асинхронно.
defer	Откладывает выполнение скрипта, пока вся страница не будет загружена полностью.
language	Устанавливает язык программирования на котором написан скрипт.
src	Адрес скрипта из внешнего файла для импорта в текущий документ.
type	Определяет тип содержимого тега <script>

<head>

```
<script src="scripts/async.js"></script>
```

</head>

Атрибуты SCRIPT, порядок выполнения

async только для внешних скриптов

Внешний Файл async.js

```
alert("I do not wait the page-parser")
```

Веб страница

```
<head>
```

```
  <script async src="async.js"></script>
```

```
</head>
```

или

```
<head>
```

```
  <script defer="defer" src="async.js"></script>
```

```
</head>
```

План лекции

1. Воспоминание. Определение Интернета и перечисление его служб. Воспоминание о протоколе HTTP.
2. Языки разметки. Понятие разметки и языков разметки. История языков разметки.
3. Структура HTML документа. Декларация типа документа, элемент `<!DOCTYPE>`, режимы браузера. Элементы `html` и `head`.
4. Некоторые тэги и атрибуты. Блочные и строчные элементы.

Блочные и строчные элементы

`<div>`

`<p>` Просто текст первого абзаца `</p>`

`<p>` Текст второго абзаца `</p>`

`</div>`

`<div>`

``

``Соседний документ``

`</p>` Текст абзаца и ``кусочек``
текста внутри абзаца`</p>`

`</div>`

Таблицы

```
<table border="1" width="400px">  
  <tr>  
    <td>Ячейка 11 </td>  
    <td>Ячейка 12 </td>  
    <td>Ячейка 13 </td>  
  </tr>  
  <tr>  
    <td>Ячейка 21 </td>  
    <td>Ячейка 22 </td>  
    <td>Ячейка 23 </td>  
  </tr>  
</table>
```

Ячейка 11	Ячейка 12	Ячейка 12
Ячейка 21	Ячейка 22	Ячейка 23

Формы

```
<form action="login.php" method="get">
```

```
<input type="text" name="username">
```

Введите имя

```
<input type="password" name="pswd">
```

Введите пароль

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

```
</form>
```

Контрольные вопросы

1. Рассмотрим HTML документ, содержащий, например,
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

а) Зачем браузеру нужна эта строка кода?

б) Здесь прописан URL адрес, откуда можно загрузить DTD. Загружает ли браузер DTD из Интернета? А если доступа к Сети нет, откуда браузер берет DTD?

2. Иногда полезно добавлять в HTML тэги "пользовательские" атрибуты. Это приводит к невалидному HTML4 коду. Пример технологии, которая приводит к невалидному HTML, но имеет огромное значение, — это ARIA. В чем суть этой технологии, для чего она нужна?

<http://www.w3.org/WAI/intro/aria>

Есть ли возможность определять валидные "пользовательские" атрибуты в HTML5?
Если да, то какая?

3. Что проверяет валидатор (X)HTML кода, чем руководствуется? Для чего нужна валидация HTML кода? Приведите аргументы "за" и "против" валидного кода?