

**ФГАОУ ВПО «Российский государственный
профессионально-
педагогический университет»**

«Web-дизайн. Основные понятия»

**Преподаватель: Власова Наталья Сергеевна,
к.п.н., доцент каф. ИС**

Екатеринбург 2015

1. Определение web-дизайна

Web-дизайн – проектная деятельность по созданию средствами web-технологий информационной архитектуры web-сайта с одновременным его художественным оформлением и оптимизацией программного кода в целях наиболее эффективного использования и продвижения в Интернет.

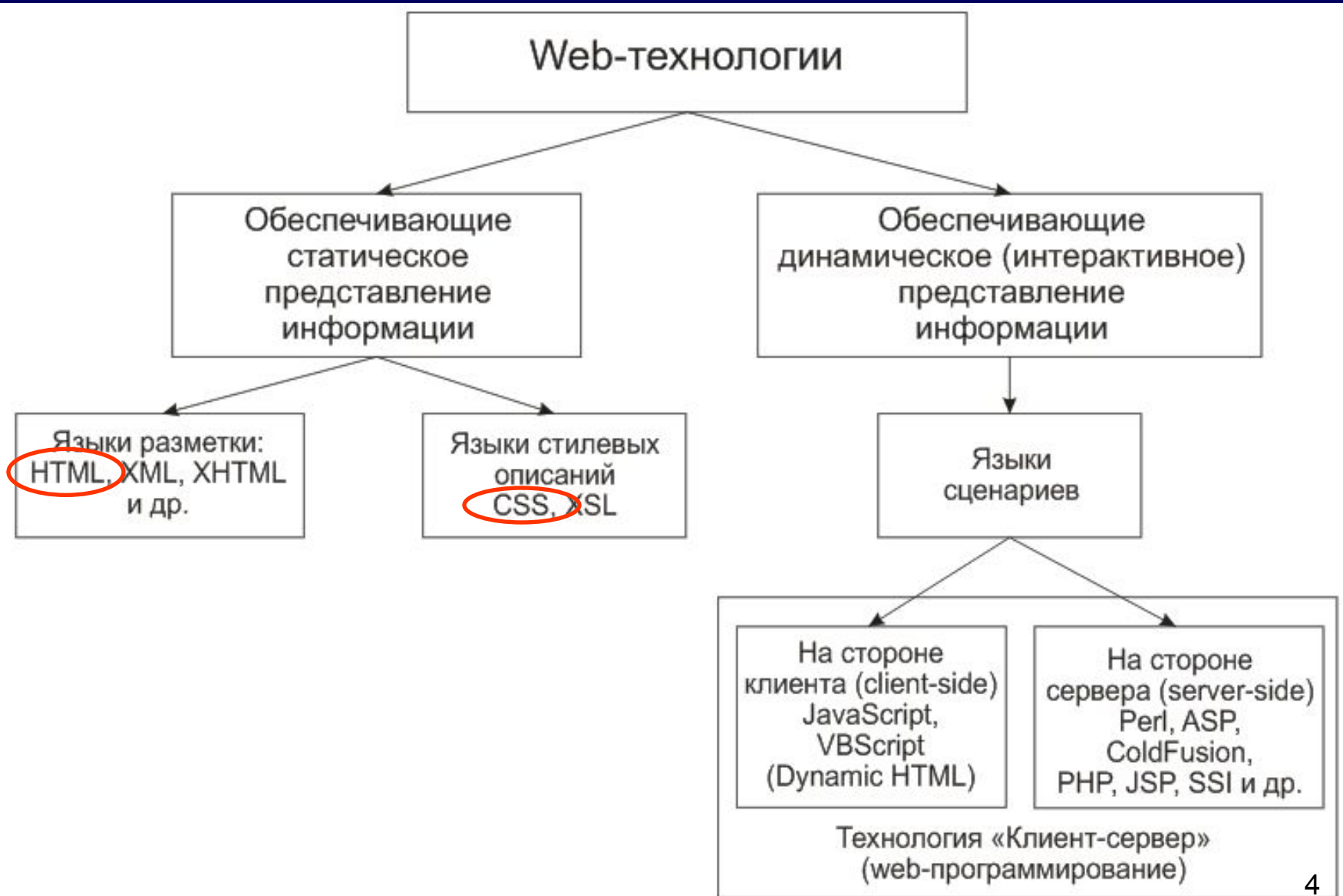
Web-сайт (сокращенно сайт) – совокупность электронных документов частного лица или организации, расположенная в сети Интернет и объединенная под одним доменным именем или IP-адресом. Все web-сайты в совокупности составляют сеть WWW.

Web-страницы сайта связаны между собой взаимными гиперссылками.

Функции web-сайта:

- информирование и привлечение пользователей;
- выполнение каких-либо операций: интернет-банк, платежные системы, интернет-магазины с регистрацией клиента и др.;
- эстетическое воздействие.

2. Технологии создания web-сайта



Статические сайты и web-страницы – это сайты, сделанные по классической технологии html. Всегда имеют один и тот же вид.

Достоинства

- простота создания;**
- нетребовательность к аппаратным ресурсам: не требуют поддержки скриптов и баз данных и могут быть размещены на абсолютно любом хостинге;**
- не создают почти никакой нагрузки на сервер;**
- могут включать в себя графику, анимацию и информацию, которая не должна часто и существенно изменяться.**

Недостаток - отсутствие интерактивности (интерактивность – взаимодействие с другими пользователями)

Динамические сайты и Web-страницы.

Не существуют в неизменном виде на сервере, а формируются при помощи скриптов.

Содержание и вид динамической страницы с одним и тем же адресом может быть совершенно различным в зависимости от разных условий – например, в зависимости от введенного пользователем запроса и т. п.

Достоинство - интерактивность:

- **проверка правильности заполнения полей;**
- **ввод запросов;**
- **поиск по сайту;**
- **отправка на сайт и сохранение своей информации;**
- **общение с другими пользователями (гостевые книги, форумы) и др.**

Недостатки

- **требуют поддержку дополнительных технологий;**
- **создают повышенную нагрузку на сервер;**
- **использование скриптов означает потенциальную угрозу для безопасности.**

Правила при создании физической (файловой) структуры сайта:

1. Имена директорий, документов HTML и графических файлов должны быть латинскими буквами и только в строчном регистре.
2. Имена должны быть «смысловыми», чтобы легче ориентироваться в содержании страниц.
3. Файл, который запускает сайт должен называться `index` (`index.html`, `index.htm`, `index.php`). Именно это имя ищет серверная машина при открытии сайта.

Простая Web-страница создается в текстовом редакторе (Notepad++ или др.) как текстовый документ и сохраняется с расширением `.html` или `.htm`.

Просматривается web-страница в браузере (Opera, Internet Explorer, Google Chrome, Mozilla Firefox и др.)

3. Инструментальные средства для создания web-сайтов

Web-редакторы:

- простые (показывают только исходный код):
NotePad и т.п.;
- WYSIWYG-редакторы (What You See Is What You Get – что видишь, то и получишь) помимо исходного кода показывают готовую web-страницу: **Microsoft FrontPage, Adobe Dreamweaver, Adobe GoLive.**
- CMS (Content Management System) – система управления контентом сайта: **Joomla!, WordPress, Drupal** и др.

- **Графические редакторы: Adobe PhotoShop, Adobe Flash, CorelDRAW и др.**
- **web-сервер Apache, Denwer, XAMPP и др.;**
 - **FTP-клиенты – для загрузки сайта на хостинг (FileZilla Client, SmartFTP, Far manager, Total Commander и др.). Многие web-редакторы имеют встроенный FTP-клиент.**
- **браузеры: Google Chrome, Opera, Mozilla Firefox, Microsoft Internet Explorer и др.**

4. Структура HTML-документа

Файлы имеют расширение **.htm** или **.html**.

Содержат один тип управляющих конструкций - **теги (tags)** – **дескрипторы**.

Все теги заключены в угловые скобки **< >**.

Обычно теги парные: **начальный** и **конечный** теги. Начальный – **<>** , конечный **</>**.

Парный тег называют еще **контейнерный**.

Некоторые теги не имеют обязательного парного тега, такой тег называют **одиночный**.

Структура

```
<!DOCTYPE ...>
```

```
<HTML>
```

```
<HEAD>..голова </HEAD>
```

```
<BODY>..тело </BODY>
```

```
</HTML>
```

```
<!-- Элемент doctype-->
```

```
<!doctype...>
```

```
<!-- Начало документа HTML-->
```

```
<html>
```

```
<!-- Начало заголовка док-та-->
```

```
<head>
```

```
<meta ... >
```

```
<title>...</title>
```

```
</head>
```

```
<!-- Конец заголовка документа-->
```

```
<!-- Начало тела документа-->
```

```
<body>
```

Текст документа.

```
</body>
```

```
</html>
```

```
<!-- Конец тела документа-->
```

Объект doctype

Является сокращением от "Document Type Definition" ("Определение типа документа"),

используется для двух целей:

- **Браузеры Web** – определяют, какой режим воспроизведения они должны использовать.
- **Валидаторы** – определяют, согласно каким правилам они должны проверять документ.

Доctype может отсутствовать.

Существует несколько видов doctype в HTML4.

Вариант Strict

Код не содержит элементов и атрибутов, помеченных W3C как «устаревшие» или «не одобряемые» (например: CENTER, FONT, U, align, background, bgcolor, color, size).

Декларация записывается в виде:

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

Вариант **Transitional**

Код может содержать устаревшие теги, вариант введен в целях совместимости со старыми версиями HTML.

Декларация записывается в виде:

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


Вариант **Frameset**

Аналогичен Transitional, но содержит также элементы для создания наборов фреймов.

Декларация записывается в виде:

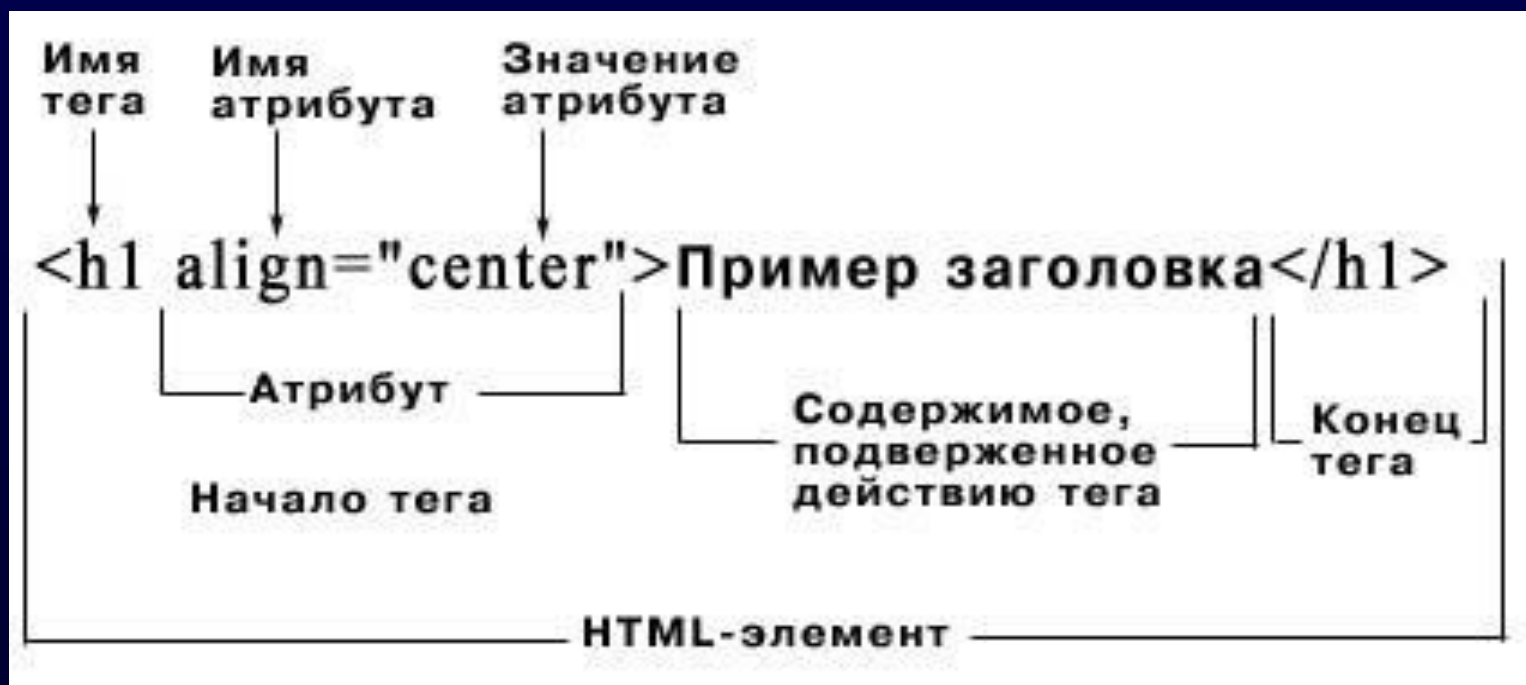
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

В HTML5 одна конструкция

```
<!DOCTYPE html>
```

5. Структура HTML-элемента (локальное форматирование)

<[имя тега] [имя атрибута 1]=[”значение атрибута 1”]
[имя атрибута 2]=[”значение атрибута 2”] ...
[имя атрибута N]=[”значение атрибута N”]>
Содержимое под действием тега</[имя тега]>



Кавычки вокруг значения атрибута обязательны в тех случаях, когда значение это содержит какие-либо символы, кроме букв, цифр, точки или дефиса, но с точки зрения совместимости с XML, лучше пользоваться кавычками всегда.

Регистр букв в идентификаторах тегов и атрибутов не учитывается.

Регистр в значениях атрибутов учитывается.

Цвет в Web

1. Система шестнадцатеричного кодирования основана на трех компонентах – красном (Red), зеленом (Green) и синем (Blue), - RGB -. от 00 до FF
2. 0 и 255 в десятичной системе счисления
3. По названию цвета

Цвет	Код	Цвет	Код
Black (черный)	#000000	Silver (серебряный)	#C0C0C0
Maroon (темно-бордовый)	#800000	Red (красный)	#FF0000
Green (зеленый)	#008000	Lime (известь)	#00FF00
Olive (оливковый)	#808000	Yellow (желтый)	#FFFF00
Navy (темно-синий)	#000080	Blue (синий)	#0000FF
Purple (фиолетовый)	#800080	Fuchsia (фуксия)	#FF00FF
Teal (темно-зеленый)	#008080	Aqua (аква)	#00FFFF
Gray (серый)	#808080	White (белый)	#FFFFFF

Значение любого цвета можно получить из
PhotoShop

Использование справочника по HTML

Тело документа

Функция	Тег	Атрибут	Значение	Примечание
Размещение основной части материала документа	<code><body>...</body></code>	Background	URL изображения	Указание URL изображения, которое используется в качестве фона
		<u>Bgcolor</u>	Цвет	Определение цвета фона документа
		<u>Leftmargin</u>	N	Ширина левого поля документа в пикселях
		<u>Topmargin</u>	N	Ширина верхнего поля документа в пикселях
		Text	Цвет	Определение цвета текста

1. Создать документ с синим цветом фона белым текстом и отступом всех элементов от верхнего края на 30 px

```
<body bgcolor=blue text=white topmargin=30>
```

Использование справочника по HTML

Форматирование текста в HTML. Основные теги

Функция	Тег	Атрибут	Значение	Примечание
Формирование заголовков	<h1>.. <h2>.. <h6>.. <h6>	Align	Left Right Center	Организация заголовков. <h1> - верхний уровень в иерархии заголовков. <h6> - нижний (самый мелкий).
Формирование абзаца	<p>	Align	Left Right Center Justify	Задаёт начало абзаца и вставляет соответствующий пустой интервал для отделения данного абзаца от предыдущего. Парный тег можно опустить
Изменение шрифта	.. 	Size	N	используется для задания абсолютного размера между 1 и 7. SIZE = +n или -n – увеличивает или уменьшает шрифт относительно текущего значения.
		<u>Color</u>		COLOR='ИМЯ' задаёт цвет текста.
		<u>Face</u>	Имя шрифта	Задаёт имя шрифта, используемого для отображения данного фрагмента текста через запятую. Использоваться будет тот, чье имя найдено первым

2. Создать заголовок второго уровня с выравниванием по левому краю красного цвета шрифтом Verdana

```
<h2 align=left><font color=red face=verdana>Заголовок 2  
</font></h2>
```

Теги, не имеющие атрибутов

Начертания шрифтов	<code>...</code> <code></code>			Текст полужирным шрифтом.
	<code><i>...</code> <code></i></code>			Текст в курсивном начертании
	<code><u>...</code> <code></u></code>			Подчеркнутый текст

Рекомендуемый ресурс <http://htmlbook.ru>

Быстрый поиск CSS или тега HTML

Перейти

Теги HTML

- <a>
- accesskey
- charset
- coords
- href
- hreflang
- media
- name
- rel
- rev
- shape
- tabindex
- target
- title
- type
- <abbr>
- <acronym>

Атрибуты тегов

Значения

14 дней FREE

Hostenko

Тег <a>

Internet Explorer	Chrome	Opera	Safari	Firefox	Android	iOS
✓ 6.0+	✓ 1.0+	✓ 2.1+	✓ 3.0+	✓ 1.0+	✓ 1.0+	✓ 1.0+

Спецификация

HTML:	3.2	4.01	5.0	XHTML:	1.0	1.1
-------	-----	------	-----	--------	-----	-----

Описание

Тег `<a>` является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов `name` или `href` тег `<a>` устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута `href` используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход. Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка. Относительные ссылки, как следует из их названия, построены относительно текущего документа или корня сайта.

Синтаксис

```
<a href="URL">...</a>
<a name="идентификатор">...</a>
```

Атрибуты

accesskey	Активация ссылки с помощью комбинации клавиш.
charset	Указывает кодировку текста, на который ведет ссылка.
coords	Устанавливает координаты активной области.
href	Задает адрес документа, на который следует перейти.
hreflang	Идентифицирует язык текста по ссылке.
media	Указывает тип носителя документа, на который ведёт ссылка.
name	Устанавливает имя якоря внутри документа.
rel	Отношения между ссылаемым и текущим документами.
rev	Отношения между текущим и ссылаемым документами.

6. Адресация в HTML

Абсолютная адресация не используется

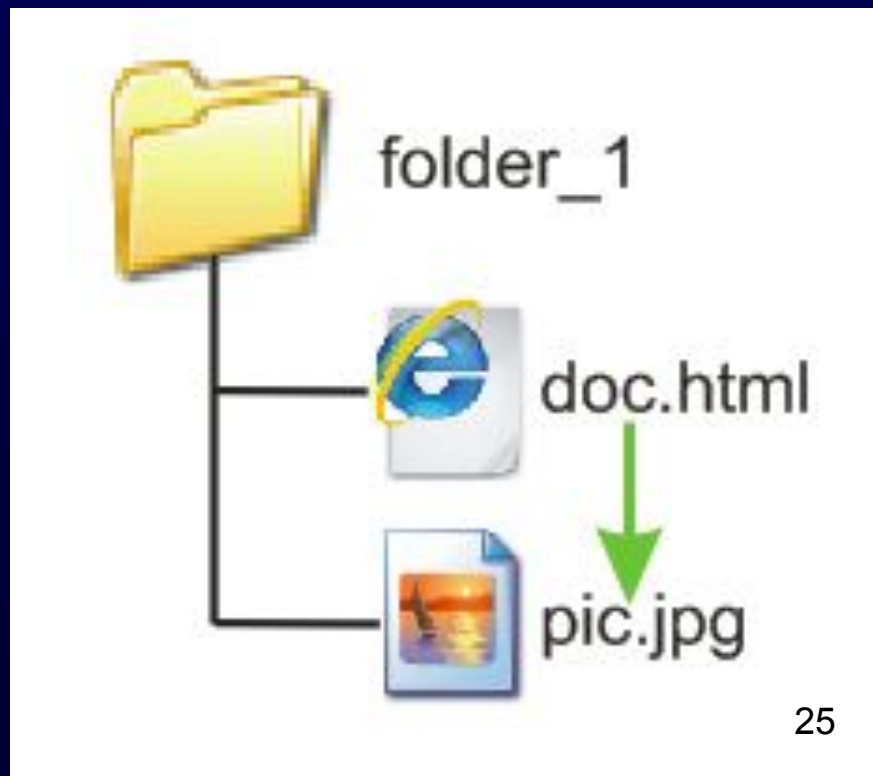
~~~~

**Используется относительная адресация**

### Пример 1

Изображение  
расположено в одной  
папке (на одном уровне) с  
HTML-документом



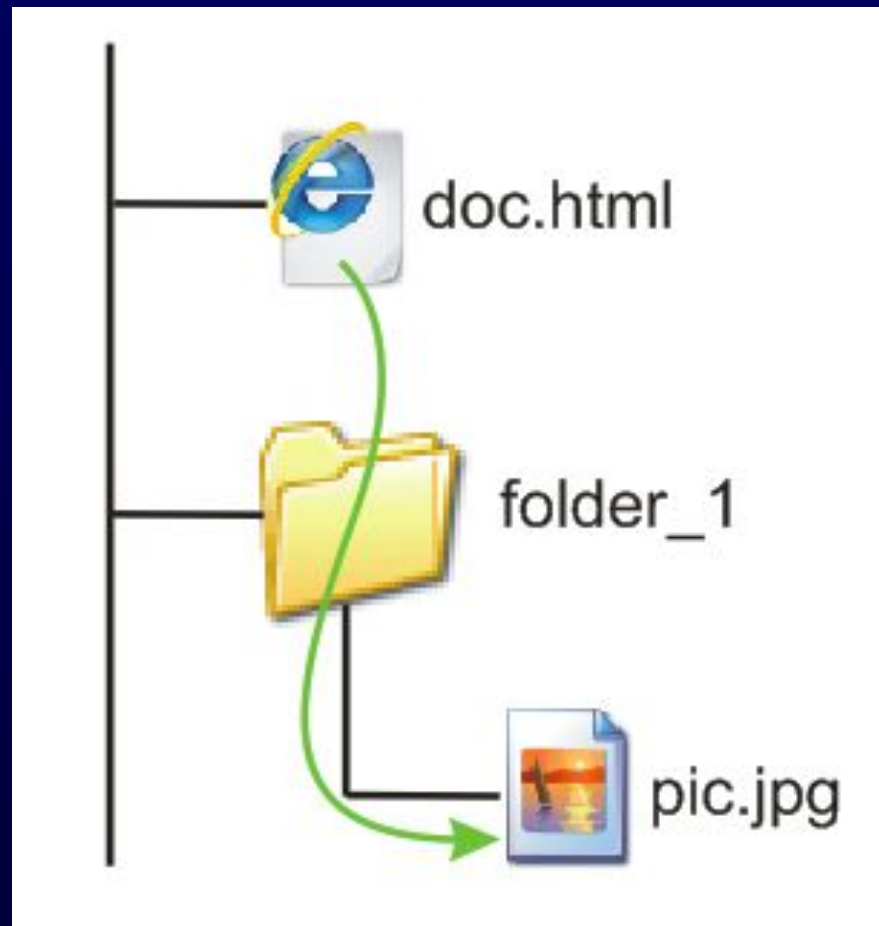


## Пример 2

HTML-документ  
расположен на один  
уровень выше, чем  
изображение

```

```

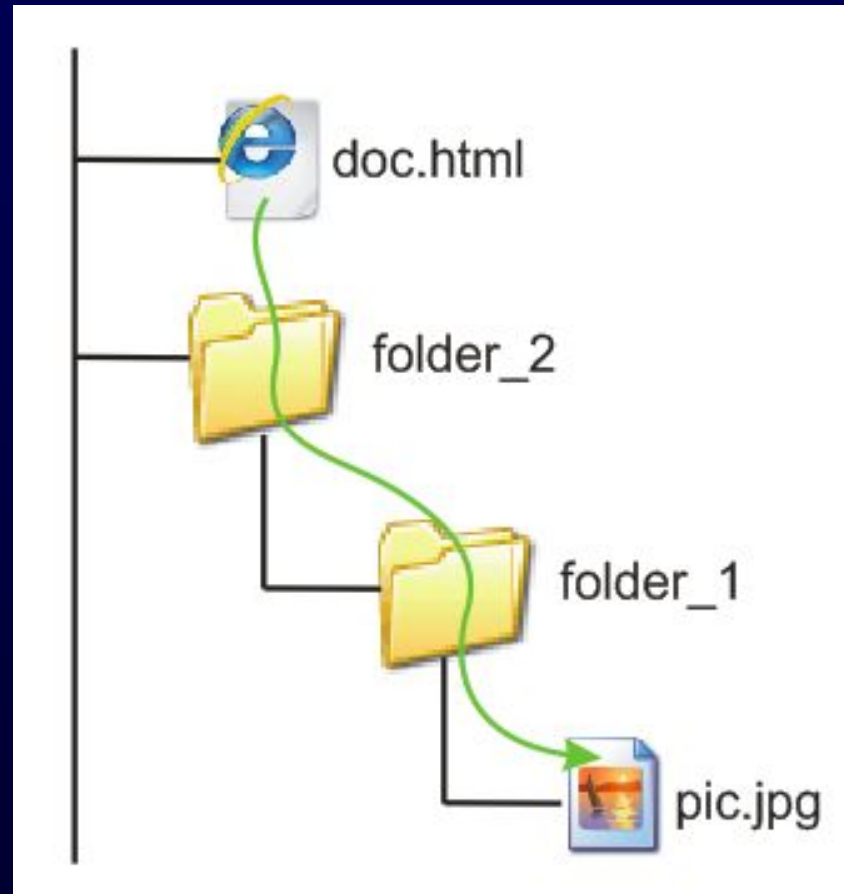


## Пример 3

HTML-документ расположен на два уровня выше,  
чем изображение

```

```

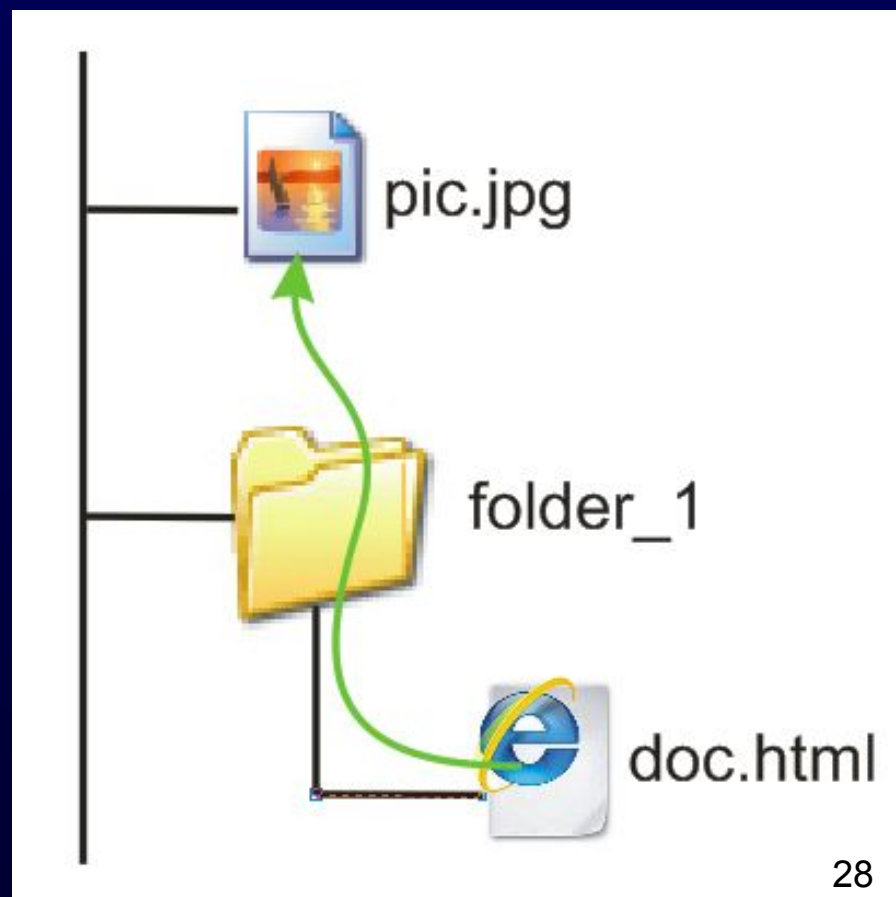


## Пример 4

HTML-документ расположен на один уровень ниже,  
чем изображение

```

```

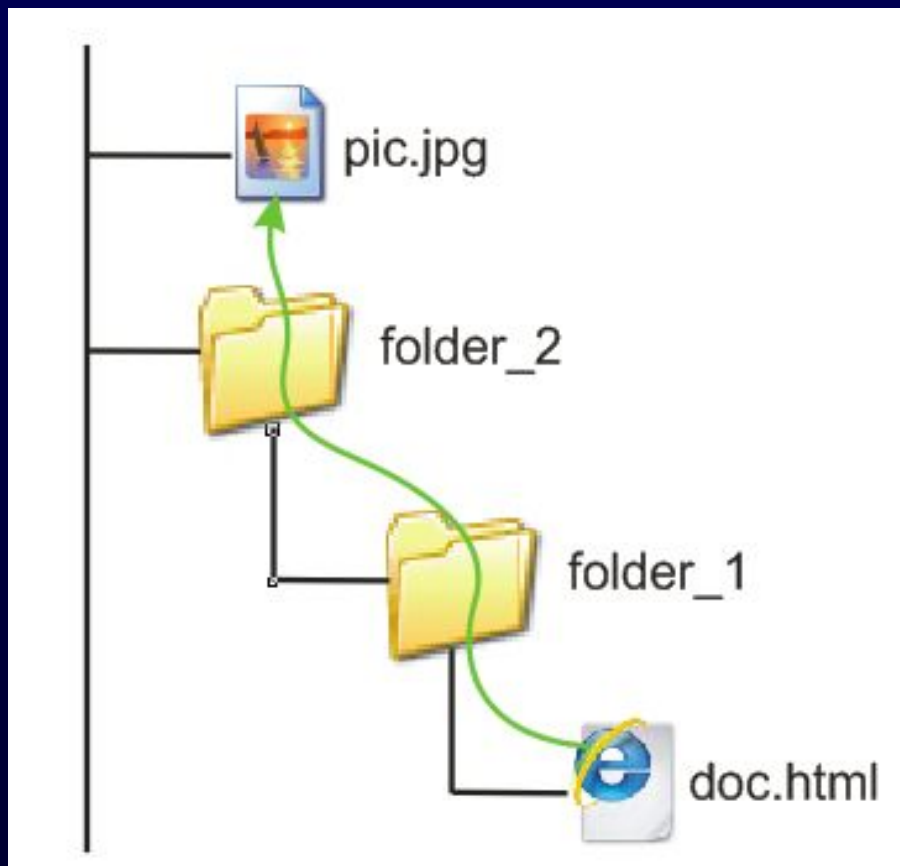


## Пример 5

HTML-документ расположен на два уровня ниже,  
чем изображение

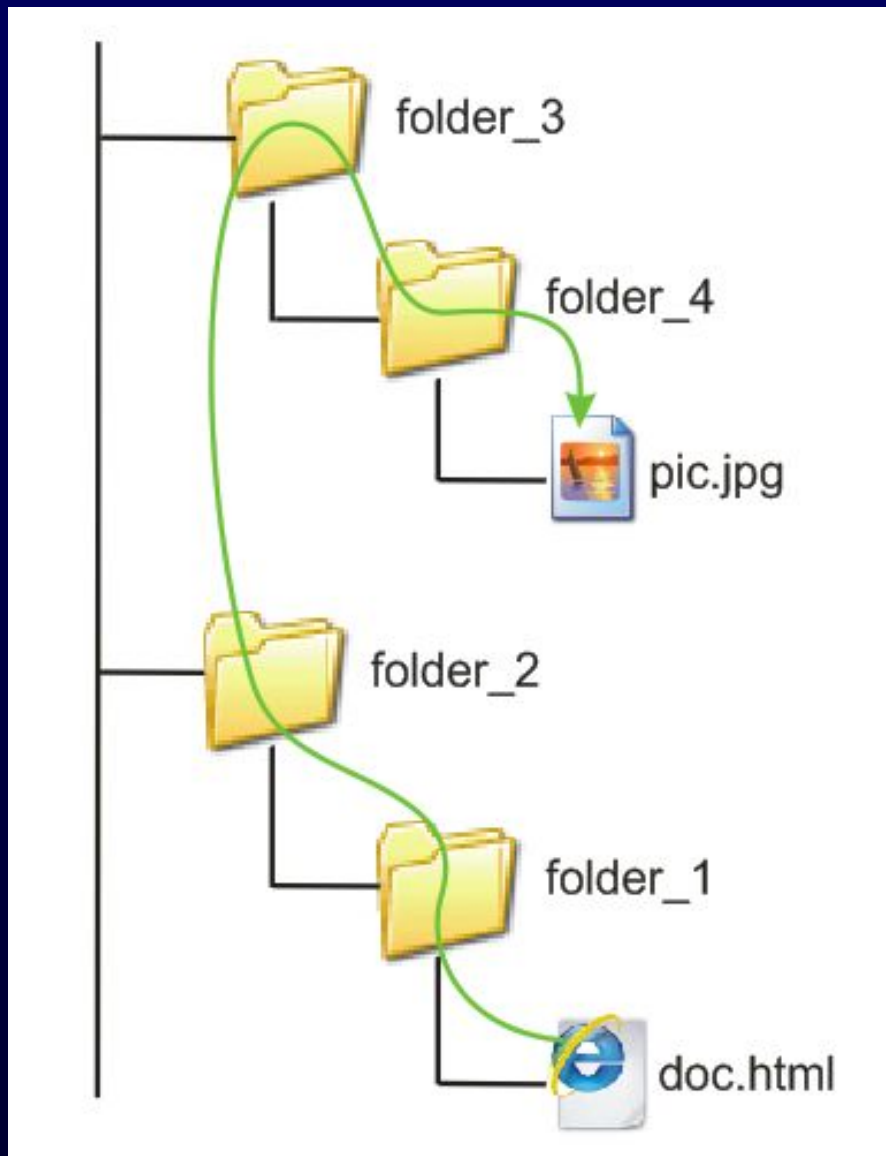
```

```



## Пример 6

``



## 7. Гиперссылки

Можно разделить на 3 категории:

1. **Внешняя** – ссылка на web-страницу, которая находится по другому адресу Internet (за пределами сайта). Для ее организации используется абсолютный адрес.
2. **Внутренняя** – ссылка на другую web-страницу внутри сайта. Для ее организации необходимо использовать относительный адрес.
3. **Внутристраничная** – ссылка на другую позицию внутри той же web-страницы.

## Структура гиперссылки

`<a href="inform.htm">здесь</a>`



Конечный якорь



Исходный якорь



## Виды адресов

**Абсолютный адрес - это полный адрес в Internet.**

- **`<a href="http://WWW.Rambler.ru">` – удаленный переход;**
- **`<a href="c:\temp\myfile.html">` – использование в ссылке абсолютного адреса на своем сервере.**

**Относительный адрес – используется для адресации в пределах документа или совокупности документов на одном сервере.**

**URL указывается относительно каталога, из которого браузер первоначально загружает Web-страницу.**

**Применение относительных ссылок удобно, т.к. при каждом перемещении каталога не приходится менять все ссылки.**

## Примеры относительной адресации

1. Переход на другой документ на одном уровне с документом, из которого делается ссылка

```
<a href="doc2.html">
```

2. Переход на документ, который находится на два уровня ниже

```
<a href="folder3/folder2/doc2.html">
```

3. Переход на три уровня вверх, затем на один уровень вниз

```
<a href="../../../folder3/doc2.html">
```

## Оглавление

`<a href=#glava1> Глава 1 </a>`

`<a href=#glava2> Глава 2 </a>`

## Заголовок статьи

`<a name=glava1> Глава 1 </a>`

Текст, текст, текст, текст, текст, текст,

`<a name=glava2> Глава 2 </a>`

Текст, текст, текст, текст, текст, текст, текст,  
текст, текст, текст

`<a href="#glava1">Глава 1</a>` – переход на конкретный фрагмент текущего документа.

Фрагмент должен быть помечен меткой с помощью тега `<a name="glava1">Глава 1</a>`

`<a href=" ../.../folder3/doc2.html#upr_20">`

## 8. Определение CSS

**CSS – Cascading Style Sheets (иерархические стилевые спецификации или каскадные таблицы стилей) не заменяют языка разметки, являются самостоятельной технологией, которая применяется к тегу HTML.**

**Назначение: позволяет автоматически изменить стиль HTML элемента на всех страницах сайта.**

**Например, мы первоначально задали на десяти web-страницах заголовкам H1 красный цвет. Затем решили изменить его на зеленый. Если H1 оформить на страницах обычным способом через тег `<font color=red>`, то нам придется десять раз изменить red на green (т.е. на зеленый). При использовании же таблицы стилей нам придется только один раз это сделать, изменив параметр red на green в самой таблице стилей.**

**Таблица стилей включает набор CSS-элементов, структура которых отличается от структуры HTML-элемента.**

## Структура CSS-элемента

селектор {стилевое свойство 1: значение;  
стилевое свойство 2: значение; ...  
стилевое свойство N: значение}

где селектор – имя HTML-элемента, атрибута CLASS, или ID),  
стилевые свойства (например, font-family)

стилевые свойства разделяются точкой с запятой.

*Например:*

```
h1 {font-family:arial; font-size:14pt}
```

# Способы включения таблиц стилей в HTML-документ

1. Внешняя стилевая спецификация
2. Внедренные стилевые таблицы
3. Внутренние стили

## Внешняя стилевая спецификация

Определяет стиль всего сайта.

Является текстовым файлом с расширением CSS.

HTML-документ должен иметь ссылку на внешнюю таблицу стилей при помощи элемента `<link>`, который располагается в элементе `<head>`:

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

## Внедренные стилевые таблицы

В заголовок документа в теге `<HEAD>` помещаются теги `<STYLE>` и `</STYLE>`, а внутри этой пары тегов вводятся стилевые правила.

```
<head>
  <style>
    Описание стиля
  </style>
</head>

<head>
  <style>
    h1 {color: blue;
        font-size:18pt;
        text-align:center}
    p {color:#0000FF;
        font-size:13pt}
  </style>
</head>
```



## Внутренние стили

Применяется к конкретному HTML-элементу внутри тега `<body>` при помощи атрибута `<style>`.

*Например:*

```
<h1 style="color:blue; font-family:arial">
```

## Классы в стилевых спецификациях

Класс определяет разновидность стиля в определенном теге, используя атрибут class.

```
<style>
```

```
  p.blue{color:blue}
```

```
  p.red{color:red}
```

```
  p.yellow{color:yellow}
```

```
</style>
```

При использовании тега `<p>` необходимо установить атрибут CLASS:

```
<p class=red> Красный текст.
```

```
<p class=blue> Синий текст.
```

Класс без указания имени элемента.

Например:

```
.my{color:#00ff00}
```

Применяется к разным элементам.

Например:

```
<p class="my">
```

```
<h2 class="my">
```

## Стиль для специфического элемента

Любому элементу можно присвоить идентификатор ID, и затем поставить в соответствие этому элементу какой-либо стиль, используя ID.

Например:

```
<style>
```

```
    #test {color:#00ffff}
```

```
</style>
```

Теперь можно поставить этот стиль в соответствие любому элементу:

```
<p ID="test">...</p>,    <h1 ID="test">...</h1>
```

## Специальные элементы

Псевдокласс – специальный селектор, определяющий, как выглядит HTML-элемент в определенный момент.

Например, псевдокласс, который устанавливает, как выглядят гиперссылки в определенные моменты времени:

`a:link {... ..}` – не посещенные ссылки;

`a:visited {... ..}` – посещенные ссылки;

`a:active {... ..}` – активные ссылки;

`a:hover {... }` – ссылка при наведении на нее фокуса.

Можно определить псевдокласс для первого символа элемента. Например, для абзаца:

`p:first-letter {font-size:24pt}`. Это удобно для задания спецэффектов выделения заглавных букв.

## Иерархия и каскадность в стилевых спецификациях

Иерархия – соподчинение.

Соподчинение таблиц стилей.

Наибольший приоритет имеет внутренний стиль (т. е. применяется в первую очередь), затем внедренный стиль (применяется во вторую очередь).

Наименьший приоритет имеет внешняя стилевая спецификация (применяется в последнюю очередь).

## **9. Установка к следующему семестру**

- 1. Контрольная работа (файл МУ\_КР\_Web-дизайн\_КТ\_испр.doc) включает одно задание – создание web-сайта средствами статических технологий HTML и CSS и написание к нему пояснительной записки.**
- 2. Обучающие материалы**
  - лабораторный практикум HTML+CSS (папка WEB\_ЛР\_2014)**

**В следующем семестре – дифференцированный зачет (зачет с оценкой).**

**Зачет состоит из следующих пунктов:**

- 1.Наличие домашней контрольной работы**
- 2.Верстка HTML шаблона на основе файла PSD (в аудитории)**
- 3.Тест по лабораторному практикуму (HTML, CSS)**



**Спасибо за внимание!**