Разработка и создание web - сайтов

Курсы ДКШ «Интернет-школа»

№ п.п.	Наименование курса	Длительность (акад.ч.)
IS-001	Основы работы в Интернет, электронная почта, создание почтового ящика, переписка	9
IS-002	Основы сайтостроения, язык гипертекстовой разметки HTML	24
IS-003	Создание веб-графики	30
IS-004	Разработка сайтов средствами Adobe Dreamweaver	12
IS-005	Проект: Разработка и создание статичного веб–сайта	15
IS-006	Создание интерактивных веб–страниц с использованием JavaScript	63
IS-007	Использование CSS, DHTML для создания веб-приложений	15
IS-008	Проект: Разработка и создание интерактивного приложения средствами JavaScript, Adobe PhotoShop, Adobe Dreamveaver	36

HTML (Hypertext Markup Language)

- это язык разметки, используемый для включения текстовых документов в Web-страницы.

Язык HTML в 1991 году создал Тим Бернерс-Ли (Tim Berners-Lee) в качестве простого способа передачи смысла и структуры гипертекстовых документов.

Роль HTML

- Говорят, что размеченный HTML-документ образует ***структурный уровень*** Web-страницы.

Это основа, над которой надстраиваются ***уровень представлений*** (инструкции по передаче и отображению элементов) и ***уровень поведения*** (скрипты и интерактивная работа).

Язык SGML (Standard Generalized Markup Language),

представляет собой сложный язык для описания структуры документов независимо от их внешнего вида.

SGML - это обширный набор правил разработки языков разметки, XHTML - это переработка HTML в соответствии с требованиями XML.

W3C

Видя необходимость упорядочить разработку HTML, Бернерс-Ли в 1994 году основал World Wide Web Consortium (W3C).

W3C продолжает надзирать за HTML и связанными с ним Web-технологиями и выпускает обновленные и стандартизованные версии HTML в виде публикаций, которые с 1995 года называются рекомендациями (Recommendations). **В настоящее время используются стандарты HTML 4.01 (1999) и XHTML 1.0 (2000).**

HTML 4.01 и XHTML 1.0

Обе рекомендации, HTML 4.01 и XHTML 1.0, включают три немного разных документа-спецификации: «Strict» (Строгий), «Transitional» (Переходный) и еще один, *Frameset DTD*, предназначенный только для документов с фреймами. В этих документах, которые называются *определениями типа документа* (Document Type Definitions, DTD) определяется каждый элемент, атрибут и сущность, а также правила их использования. DTD для XHTML написаны в соответствии с правилами и соглашениями XML (Extensible Markup Language), а DTD HTML используют синтаксис SGML. Браузер использует данные DTD для «расшифровки» разметки и проверки ее допустимости.

Элементарная структура документа

- <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
- <html xmlns="http://www.w3.org/1999/xhtml">
- <head>
- <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
- <title> Название документа</title>
- </head>

- <body>
- Содержимое документа
- </body>
- </html>

- Создавая XHTML-документы (и связанные с ними таблицы стилей), убедитесь, что все теги и имена атрибутов находятся в нижнем регистре. Для значений атрибутов требования к учету регистра отсутствуют.
- В XHTML все элементы должны иметь завершающий тег, иначе будет зарегистрирована ошибка и документ будет признан не соответствующим стандарту. Завершать необходимо и пустые элементы. (
)
- XML (а следовательно, и XHTML) не поддерживает сокращенную запись атрибутов

Перечислим атрибуты, которые являлись сокращенными в HTML, но для которых в XHTML требуются значения

- **checked=«checked»**
- **disabled=«disabled»**
- **noresize=«noresize»**
- **compact=«compact»**
- **ismap=«ismap»**
- **nowrap=«nowrap»**
- **declare=«declare»**
- **multiple=«multiple»**
- **readonly=«readonly»**
- **defer=«defer»**
- **noshade=«noshade»** **selected= «selected»**

Пустые элементы

<area /> <frame /> <link />

<base /> <hr /> <meta />

<basefont /> <img /> <param />

 <input />

<col /> <isindex />

Заголовок документа

Каждый элемент **head** должен включать в себя элемент **title**, который содержит описание документа. Элемент **head** также может включать любой из следующих элементов в любом порядке: **script, style, meta, link, object, base**. Элемент head служит просто контейнером для этих элементов и не имеет своего собственного содержимого.

Указание типа информации и кодировки символов

- Рекомендуется (хотя и не является обязательным), чтобы тип информации и кодировка символов указывались внутри документов (X)HTML, чтобы эта информация сохранялась в документе

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

content-language

Это значение может использоваться для указания языка, на котором написан документ.

В данном примере для браузера указывается, что естественным языком документа является французский:

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

Имена в элементе meta для поисковых систем

<meta name="description" content="Toropova Olga' resume and web design samples" />

<meta name="keywords" content="designer, web design, branding, logo design" />

<meta name="author" content=" Toropova Olga" />

<meta name="copyright" content="2009, O'Reilly Media" />

Тело документа

<body>...</body>

Атрибуты

Базовые атрибуты: id, class, style, title

Внутренние события: onload, onunload, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkey-down, onkeyup

Устаревшие атрибуты

alink="#rrggbb" или "имя цвета"

background="URL"

bgcolor="#rrggbb" или "имя цвета"

link="#rrggbb" или "имя цвета"

text="#rrggbb" или "имя цвета"

vlink="#rrggbb" или "имя цвета"

Текстовые элементы делятся на две главные категории: **внутристрочные (inline) и блочные.**

Внутристрочных элементы встречаются в потоке текста и по умолчанию не приводят к переводу строки.

Блочные элементы по умолчанию начинаются в представлении с новой строки и состыковываются в обычном потоке документа как блоки.

Блоковые элементы

h1-h6	Заголовок
p	Абзац
pre	Предварительно форматированный текст
address	Контактная информация
blockquote	Длинная цитата

Внутристрочные элементы

- **em** – обозначает выделенный текст. Элементы **em** почти всегда отображаются курсивом.
- **strong** – обозначает сильно выделенный текст. Элементы **strong** почти всегда отображаются полужирным текстом.
- **abbr** – обозначает сокращенную форму.
- **acronym** – обозначает аббревиатуру.
- **cite** – обозначает цитату - ссылку на другой документ, в особенности на книги, журналы, статьи и т. п. Эти элементы обычно отображаются курсивом.
- **q** – **о**бозначает краткую внутристрочную цитату.
- **dfn** – обозначает определение или первое вхождение заключенного в тег термина. Может использоваться для привлечения внимания к появлению специальных терминов и фраз. Определения терминов часто отображаются курсивом.

Внутристрочные элементы

- **code** – обозначает фрагмент программного кода. По умолчанию код отображается в браузере специальным шрифтом фиксированной ширины (обычно - Courier).
- **kbd** – является сокращением от слова «keyboard» (клавиатура) и обозначает текст, введенный пользователем. Как правило, такой текст отображается шрифтом фиксированной ширины.
- **samp** – обозначает пример выходных данных программы, скрипта и т. п. Как правило, такой текст по умолчанию отображается шрифтом фиксированной ширины.
- **var** – обозначает экземпляр переменной или аргумент программы. Переменные обычно отображаются курсивом.
- **sub -** Подстрочный
- **sup -** Надстрочный

br Перенос строки

Обозначение редактирования

ins Вставленный текст

del Удаленный текст

Общие элементы

Общие элементы **div** и **span** предоставляют авторам возможность создавать собственные элементы. Элемент **div** применяется для обозначения блочных элементов, а элемент **span** - для обозначения внутристрочных элементов.

div Блок

span Участок внутристрочного текста

Стилевые элементы

b	Полужирный
big	Текст большого размера
i	Курсив
s	Зачеркнутый
strike	Зачеркнутый
tt	Телетайп
u	Подчеркнутый
font	Гарнитура шрифта, цвет и размер
basefont	Задает размер шрифта по умолчанию
nobr	Нет переноса строки
wbr	Перенос слова
hr	Горизонтальная линейка

Списки

- Несортированная информация (маркированный).
- Сортированная информация (нумерованный).
- Термины и определения

Несортированные списки

В (X)HTML неупорядоченные списки обозначаются элементом ul. Содержимое элемента ul ограничивается одним или несколькими пунктами списка (элементы li).

Несортированные списки, а также их пункты представляют собой блочные элементы, так что каждый из них отображается начиная с новой строки.

ul

...

Атрибуты

Базовые (id, class, style, title)

Устаревшие атрибуты

type="**disc** | **circle** | **square**"

li

...
Синтаксис несортированных списков

Пример

```
<ul>
    <li>Unordered information</li>
    <li>Ordered information</li>
    <li>Terms and definitions</li>
</ul>
```

Атрибуты

Базовые (id, class, style, title)

Устаревшие атрибуты

type=«format»

value="number"

Сортированные списки

Сортированные списки обозначаются элементом ol, и они должны включать в себя один или несколько пунктов (элементов li). Как и все прочие списки, сортированные списки и их пункты представляют собой блочные элементы.

- **ol**

..,

Атрибуты

Базовые (id, class, style, title)

Устаревшие атрибуты

start="number"

type= “1| l | A | a | i "

Сортированные списки имеют ту же базовую структуру, что и несортированные, и это показано в следующем простом примере.

```
<ol>
<li>Встать с кровати</li>
<li>Принять душ</li>
<li>Погулять с собакой</li>
</ol>
```

Списки определений

- Используйте *список определений* для тех списков, которые состоят из пар термин - определение.
- Списки определений обозначаются элементами dl. Содержимое элемента dl представляет собой некоторое количество терминов (обозначенных элементом dt) и определений (обозначенных элементом dd). Элемент dt (термин) может содержать только внутристрочные материалы, но элемент dd может включать блочные или внутристроч-ные элементы. Все три элемента, используемые в списках определений, представляют собой блочные элементы, и они по умолчанию начинаются с новой строки.
- dl
- **<dl>...</dl>**
- **Атрибуты**
- *Базовые* (id, class, style, title), *интернационализация*, *события*
- compact dd
- **<dd>...</dd>**

. <dl>
<dt>**em**</dt>
<dd>обозначает выделенный текст. Элементы em почти всегда отображаются курсивом.
</dd>
<dt>strong</dt>
<dd> обозначает сильно выделенный текст. Элементы Strong почти всегда отображаются жирным шрифтом.
</dd>
<dt>abbr</dt>
<dd> обозначает сокращенную форму.</dd>
<dt>acronym</dt>
<dd> обозначает аббревиатуру.</dd>
</dl>

СТИЛИ CSS

- Стиль – всё то, что определяет внешний вид Web-документа при его отображении в окне браузера.
- Главная идея каскадных таблиц стилей – разделить содержимое документа и его физическое представление.Таблицы стилей (так называемые **Cascading Style Sheets** – CSS) содержат описание формата части или всего текста, координаты расположения элементов и другие параметры.

Преимущества CSS

- *Больший контроль над шрифтовым оформлением и раскладкой страницы*
- *Потенциальное уменьшение размера документов*
- *Потенциальное увеличение доступности документа*
- *Стилевой HTML выходит из употребления*
- *CSS хорошо поддерживается*

История CSS и разработка стандартов

Никогда не предполагалось, что **HTML** будет языком представления, так что мысль об отделении таблиц стилей от HTML-документов существует с 1990 года, когда Web была лишь идеей в голове Тима Бернерса-Ли

.

В 1995 г., когда начал свою работу консорциум **World Wide Web Consortium** (W3C), была создана официальная рабочая группа, занимающаяся CSS. К этому моменту из названия таблиц стилей исчезло слово «HTML», поскольку сразу стало понятно, что другим языкам разметки также требуется язык для представления.

Первой официальной рекомендацией CSS стала Рекомендация **CSS Level 1**, выпущенная в 1996 году, которая содержит все основные механизмы применения инструкций, связанных со шрифтами, цветами и интервалами к элементам страницы.

Рекомендация **CSS Level 2** была выпущена в 1998 году. Наиболее заметными нововведениями в ней стали свойства для позиционирования элементов на странице (которые исходно были выпущены под названием **CSS-P**, позже превратившееся в **CSS Level 2**), но, кроме того, среди прочего были предложены типы данных, свойства для табличных раскладок, акустические таблицы стилей и более сложные методы выбора элементов.

В разработке находятся еще две рекомендации. **CSS Level 2, Revision 1**, который добавляет небольшие корректировки к CSS2 на основе опыта, полученного за период с 1998-го по 2004 год.

Исправлены ошибки, были удалены свойства, не принятые CSS-сообществом, а некоторые неподдерживаемые свойства перенесены в будущую спецификацию **CSS 3**.

В модульной Рекомендация **CSS Level 3** добавляется поддержка вертикального потока текста, усовершенствованная обработка таблиц, поддержка разных языков и более совершенная интеграция с другими XML-технологиями, такими, как SVG (Scalable Vector Graphics), MathML и SMIL (Synchronized Multimedia Interchange Language).

Чтобы познакомиться с самой последней информацией о деятельности W3C в области CSS, обращайтесь на сайт ***www.w3.org/Style/CSS/***.

- задание правила состоит из 2-х частей: ***селектора*** и ***определения***.
- **Селектор** – имя любого тега HTML, класса, или идентификатор.
- **Определение** – задают свойства селектора.
- Внешний вид списка стиля:
- **Селектор {свойство: значение;**
- **свойство: значение;}**

```
<HTML>
  <HEAD>
  <STYLE>
H1 {font-size: 48pt;
      color:RED}
H2 {font-size: 20pt;
      color:BLUE}
   </STYLE>
   </HEAD>
<BODY>
          <H1>Это стиль H1. Цвет - красный</H1>
          <H2>Это стиль H2. Цвет - синий</H2>
<P>      Это - обычный стиль по умолчанию  </P>
</BODY>
</HTML>
```

Способы добавления таблиц стилей на Web-страницы

- **Встраивание стилей** (внутрь тега)
- <p style = “color: red”>
- <p style = “color: red; font-size:60pt”>
- **Включение стилей** (заголовочные стили).
- В область заголовка добавить стиль STYLE
- **<head>**
- **<style type=”text/CSS”>**
- **P {color: blue;**
- ** background-color: yellow}**
- **</style>**
- **</head >**

Связывание

Таблицу стилей создают в виде отдельного текстового файла с расширением **.css**. Документ таблицы стилей – это текстовый документ, который содержит как минимум одно стилевое правило. Он не может содержать тегов HTML.

Этот стиль можно применить к нескольким документам.

Пример. ***В файле C1.CSS пишем следующий текст:***

```
P     {color: green;
       font-size: 30 pt}
body {background-color: pink}
```

в документе: **<head>**

<link rel = “stylesheet” type = “text/css” href = “c1.css”>

… </head>

Тег **LINK** позволяет определить, на какой внешний документ мы будем ссылаться.

REL определяет тип связи между текущей страницей и той, на которую указывает ссылка. Если REL = stylesheet – мы связываемся с таблицей стилей.

TYPE – определяет тип того документа, с которым связываемся.

HREF – позволяет задать URL-адрес таблицы стилей.

Импортирование

- позволяет встраивать в документ таблицу стилей, расположенную на сервере.
- Для этого в область заголовка добавляют следующий код:
- **<style type=”text/CSS”>**
- **@ import url (**http://www.myserver.ru/CSS/line.CSS**)**
- **</style>**
- Команды **@import** должны идти перед всеми остальными элементами (за исключением **@charset**).
- Импортирование позволяет применить несколько таблиц стилей к одному документу. Если в элемент **style** добавляются дополнительные функции **@import**, то информация из файла, прочитанного последним, имеет преимущество перед предыдущими.
- Директива **@import** может также использоваться в самой таблице стилей для ссылки на информацию во внешних CSS-файлах.

Единицы измерения, используемые в каскадных таблицах стилей

in – дюйм = 2, 54 см

- **cm – сантиметры**
- **mm – миллиметры**
- **px – пиксели**
- **pt – пункты = 1/72 дюйма ~ 0,375 мм**
- **pc = пика = 12 pt**
- **% – процент**
- **em** – относительная единица измерения, которая обычно равна ширине заглавной буквы «M» в текущем шрифте. В CSS эта единица равна размеру шрифта в пунктах (т. е. ширина em в шрифте 24pt равна 24 пунктам) и используется для указания размера как по горизонтали, так и по вертикали
- **ex** – Относительная единица измерения, равная высоте буквы «x» в нижнем регистре для текущего шрифта (приблизительно половина em)

Свойства CSS для управления цветом

- Цвет можно задать:
- названием; (red, blue, ...)
- кодом: #RRGGBB
- кодом #RGB
- В данном методе используется трехзначный синтаксис, который преобразуется в шестизначную форму путем повторения каждой цифры (следовательно, #00F аналогично #0000FF).
- с помощью функции: rgb (R, G, B), например color:rgb (0, 255, 0);
- rgb (R%; G%; B%)
- color:rgb(0%; 0%; 100%)

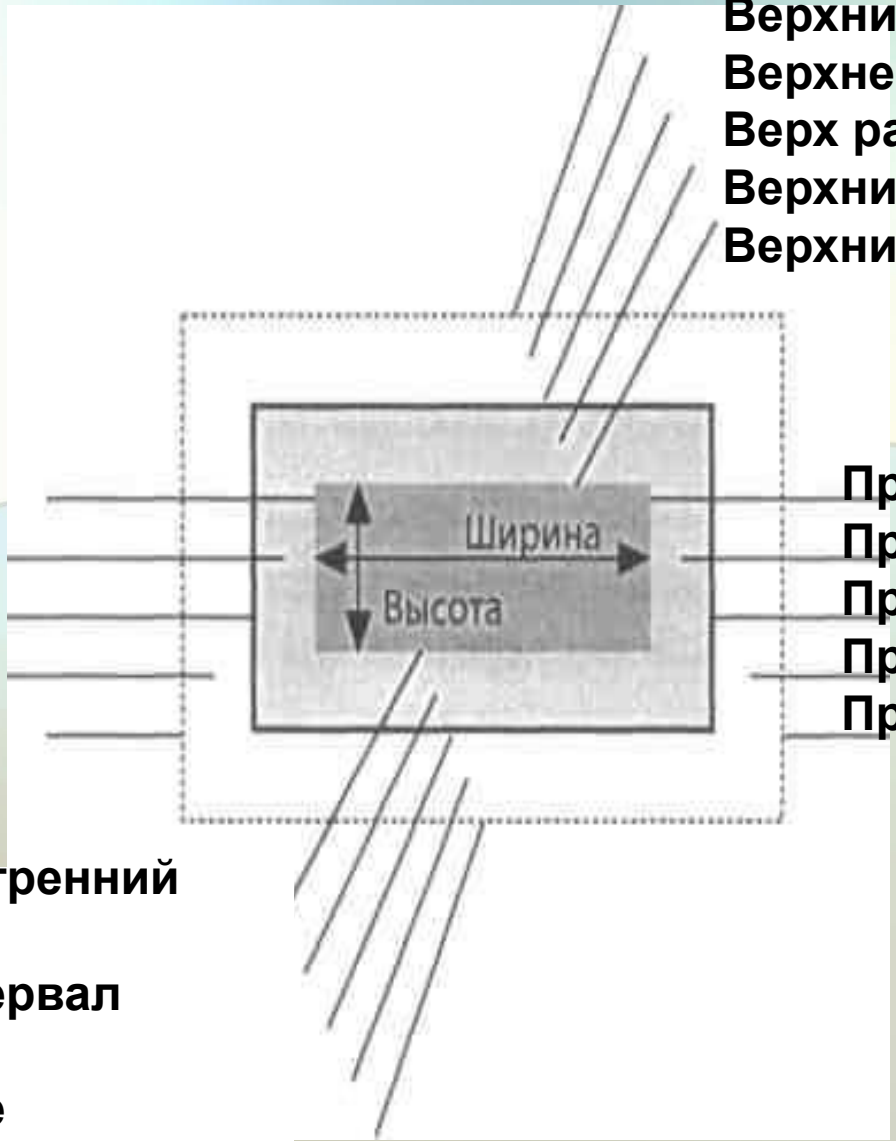
Конфликтующие правила стилей: каскад

Типичной является ситуация, когда элементы документа получают инструкции по представлению из нескольких источников. При этом обязательно возникают конфликты. Рабочая группа, создававшая CSS, предвидела эту ситуацию и разработала иерархическую систему, в которой различным источниками информации о стилях присваиваются разные веса.

Каскадом (в каскадной таблице стилей) называется то, что происходит, когда несколько источников информации о стилях соперничают за контроль над элементами страницы. Информация о стилях передается ниже по иерархии, пока не будет перезаписана информацией, имеющей больший **вес**.

В каскадном порядке существует ряд правил для разрешения конфликтов между конкурирующими таблицами стилей. Когда пользовательский агент (например, браузер) встречает элемент в коде документа, он просматривает все объявления стилей, которые могут к нему применяться, а затем сортирует их по их происхождению таблицы стилей, специфичности селектора и порядку правил и на этом основании определяет, какая таблица применяется.

Верхний наружный край
Верхнее поле
Верх рамки
Верхний интервал
Верхний внутренний край



Левый внутренний край
Левый интервал
Левая сторона рамки
Лев
Лев
нару

Правый внутренний край
Правый интервал
Правая сторона рамки
Правое поле
Правый наружный край

Нижний внутренний край
Нижний интервал
Низ рамки
Нижнее поле
Нижний наружный край

Основные свойства блока, предназначенные для указания размера, добавления полей, границ и интервала.

height
width
max-height
max-width
min-height
min-width
border-top-width
padding-bottom
padding-left
padding
padding-top
padding-right

border-top-style
border-right-style
border-bottom-style
border-left-style
border-style
border-right-width
border-top
border-right
border-bottom
border-left
border
border-bottom-width
border-left-width
border-width

margin-left
margin-top
margin-bottom
margin-right
margin
border-top-color
border-right-color
border-bottom-color
border-left-color
border-color

Свободное размещение и позиционирование

Спецификации CSS не сводятся лишь к «украшательству» элементов, образующих документ. Их также можно использовать для задания **базовой структуры страницы**. Познакомимся с двумя принятыми в CSS методами упорядочения элементов: *свободным размещением (**float-ing**) и позиционированием (**positioning**).*

Перечислим свойства, управляющие положением элементов в CSS 2.1:

float	**bottom**	**overflow**
clear	**top**	**clip**
position	**left**	**visibility**
bottom	**right**	**z-index**

Нормальный поток

В модели «нормальный поток» (**normal flow**) текстовые элементы страницы располагаются сверху вниз и слева направо для языков, в которых читают слева.

Входя в **нормальный поток**, блочные элементы располагаются друг за другом, а встроенные – заполняют доступное для них место.

При изменении окна браузера блочный элемент расширяется или сужается, а содержимое элемента форматируется с учетом изменившейся ширины. В этой модели объект влияет на положение соседних, окружающих элементов.

Свободное размещение

В CSS можно свободно расположить любой мыслимый элемент (абзац, список, элемент div и т. д.), а не только изображение.

Свободным размещением элементов пользуются для построения документов с несколькими колонками; панелей навигации из ненумерованных списков; выравнивания, напоминающего таблицу, но не являющуюся таковой, и т.д.

Для размещения элемента у левой или правой границы и обеспечения обтекания текста вокруг него используется свойство

float.

Значения: **left** **right** **none** **inherit.**

Начальное значение: none

Применимо: Ко всем элементам.

В этом простом примере свойство float используется, чтобы прижать изображение вправо

```
img   {float : right;
          margin : 20px;}
```

Пропуск размещаемых элементов

Бывает, что область сбоку от свободно размещенного элемента хочется оставить незанятой, а следующий элемент начать в «нормальной» позиции. В этих случаях, чтобы помешать появлению элемента рядом со свободно размещенным объектом, используйте свойство **clear** со значениями **left, right, both, none, inherit**. Начальное значение: **none**.

Данное свойство применимо к элементам уровня блока.

```
img  {float:  left;
        margin-right:  10px;  }
h1  {clear:  left;
      top-margin:  2em; }
```

Аналогично работает и значение **right**, предотвращающее появление элемента рядом с размещенными справа. Значение **both** смещает элемент вниз, пока не будут пропущены свободно размещенные элементы справа и слева.

Позиционирование: основы

Существует четыре типа позиционирования, каждый из которых задается свойством position со значениями **static, relative, absolute, fixed, inherit.**

Начальное значение: **static**

Свойство **position** указывает, что положение элемента должно быть подобрано специально.

- **static** Обычная схема позиционирования, в которой прямоугольники элементов отображаются на экран в порядке их появления в документе.
- **relative** Относительное позиционирование перемещает прямоугольник, где содержится элемент, однако исходное пространство под него в документе сохраняется за самим элементом.
- **absolute** Объекты с абсолютной позицией изымаются из потока содержимого документа и размещаются относительно охватывающего блока. В силу удаления из потока элемент больше не влияет на размещение окружающих элементов.
- **fixed** Выбор фиксированной позиции подобен выбору абсолютной (элемент удаляется из потока содержимого документа), однако положение элемента определяется относительно порта просмотра (в большинстве случаев окна браузера), а не того элемента, который содержит данный.

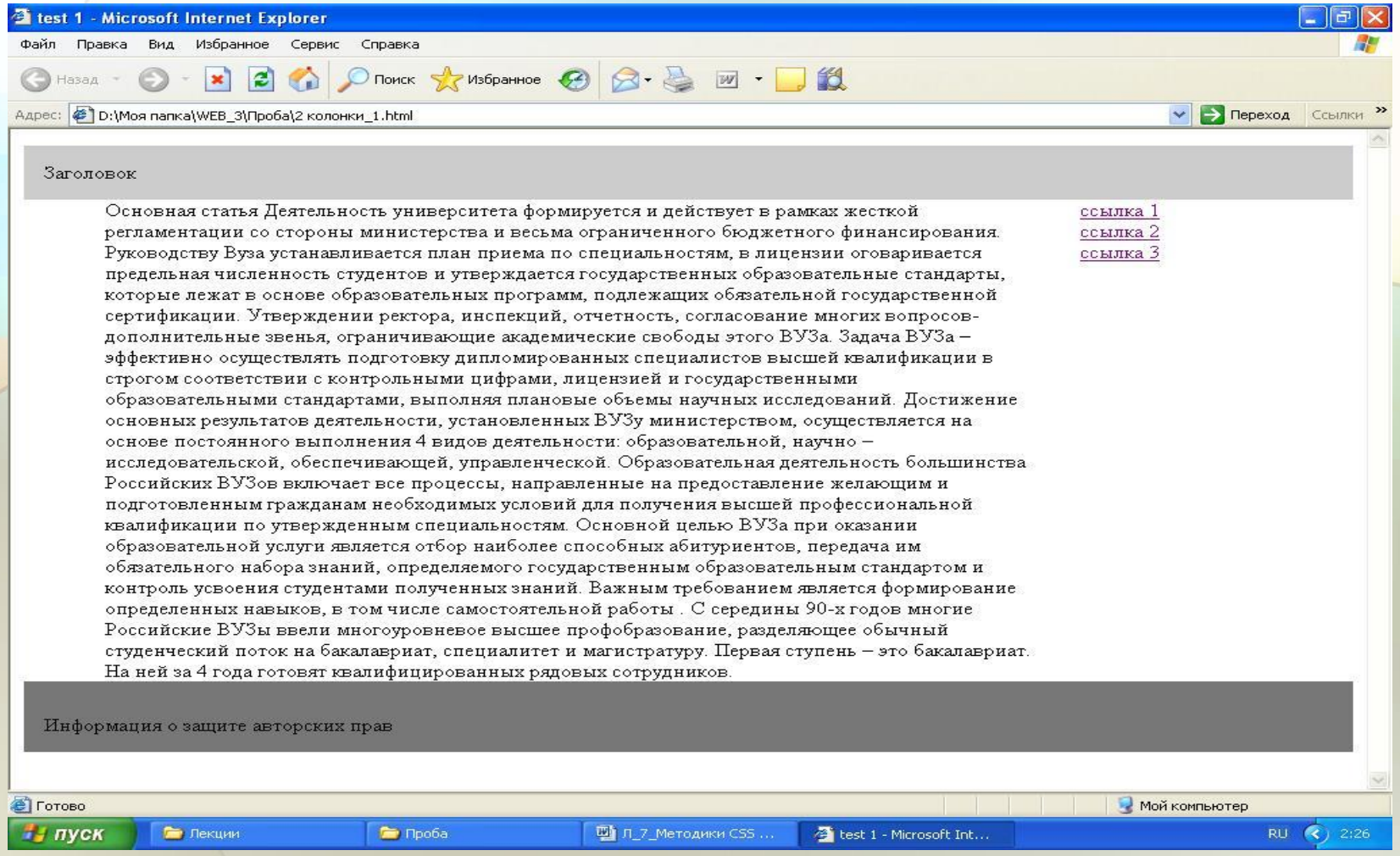
Методики CSS

- **Центрирование страницы**
- 1 способ
- **#page { width:500px;**
- **margin-left:auto;**
- **margin-right:auto;**
- **}**
- 2 способ
- **body {text-align:center;}**
- **body * {text-align:left}**
- **#page { width:500px;**
- **margin:0 auto;**
- **}**

3 способ

#page { position:absolute;
left:50%;
width:500px;
margin-left: -250px;

Раскладка в 2 столбца



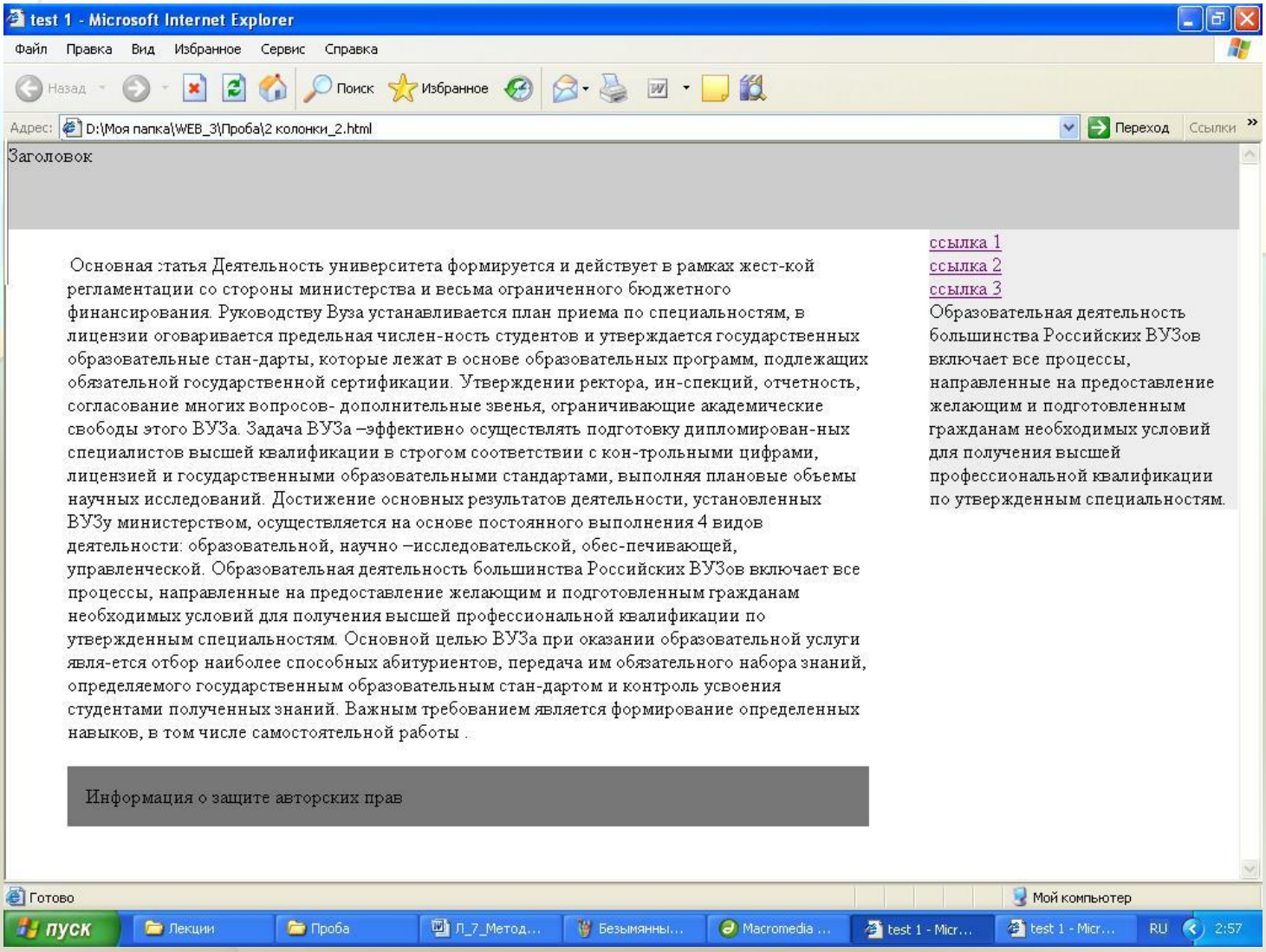
Раскладка в 2 столбца

- Использование плавающих элементов
- <body>
-
- Шапка и осн Заголовок
-
-
- Основная статья
-
-
- ссылки
-
-
- Информация о защите авторских прав
-
- </body>

Таблица стилей

```
.masthead {background:#ccc;
           padding:15px;
          }
.main       {float:left;
           width:70%;
            margin-right:3%;
            margin-left:3%;
          }
.footer {clear:left;
 padding:15px;
background:#777;
           }
```

Использование абсолютного позиционирования



Использование абсолютного позиционирования

```
body {margin:0; /* убираем пространство по умолчанию вокруг страницы*/
    padding:0;
    }
.masthead {background:#ccc;
            height:70px;
            }
.main {    margin-right:30%;/* место для бокового столбца*/
        margin-left:5%;
        }
  }
```

```
.links{         position:absolute;
        top:70px;
        right:0px;
        width:25%;
        background:#eee;}

.footer {  padding:15px;
           background:#777;
           margin-right:30%;
           margin-left:5%;
```

- Предлагаю посмотреть работы слушателей Детской компьютерной школы!

Спасибо за внимание!