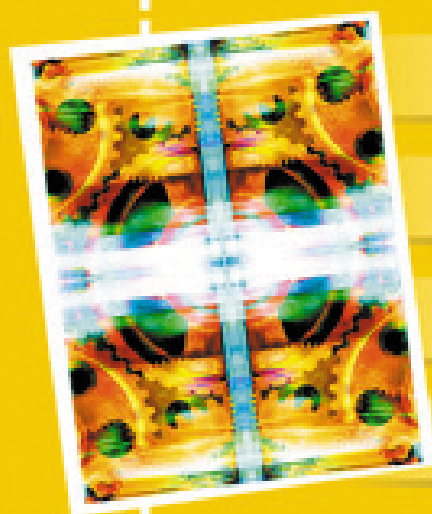


WEB-КОНСТРУИРОВАНИЕ. DHTML



Правила построения
каскадных стилевых таблиц

Основы языка
сценариев JavaScript

Объектная модель
гипертекстового документа

Практика создания
"живых" сайтов

О С Н О В Ы

И Н Ф О Р М А Т И К И

А. А. Дуванов

WEB-КОНСТРУИРОВАНИЕ. DHTML

Санкт-Петербург
«БХВ-Петербург»

2003

УДК 681.3.06
ББК 32.973.26-018.2
Д79

Дуванов А. А.

Д79 Web-конструирование. DHTML. — СПб.: БХВ-Петербург, 2003. — 512 с.:
ил.

ISBN 5-94157-334-0

Это вторая книга серии "Web-конструирование". Она пополняет навыки создания гипертекстовых страниц на базе "чистого" HTML правилами построения каскадных стилевых таблиц (CSS) и основами программирования сценариев на языке JavaScript.

Подробно рассматривается объектная модель браузера, на конкретных примерах объясняется методика построения собственных программных объектов, интегрируемых в общую среду "браузер-гипертекстовое приложение". Большое внимание уделяется структуре программного кода, дисциплине программирования, тестированию и отладке сценариев.

Для учащихся 10—11 классов и преподавателей школ

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Елена Михальчук</i>
Компьютерная верстка	<i>Татьяны Олоновой</i>
Корректор	<i>Вера Александрова</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.06.03.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 41,28.

Тираж экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953 Д.001537.03.02
от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов
в Академической типографии "Наука" РАН
199034, Санкт-Петербург, 9 линия, 12.

Содержание

Предисловие	1
Напутствие читателю	1
Гипертекстовые учебники Роботландского университета	2
Поддержка книги (коды примеров)	2
Инструментарий занятий	3
Устройство книги	4
Благодарности	6
Введение	7
Что такое DHTML?	7
Что такое CSS?	8
ЧАСТЬ I. ОСНОВЫ CSS	11
Урок 1. CSS в HTML-коде	13
Стиль для отдельного тега	13
Стиль для отдельного HTML-файла	14
Стиль для нескольких HTML-файлов	17
Комбинирование стилей	17
За что мы любим CSS?	21
Зачетный класс	21
Задания	22
Урок 2. Обзор свойств CSS	24
Единицы измерения	24
Шрифт	25
Цвет	28
Текст	29
Поля и рамки	31
Вид	32

Зачетный класс	33
Задания	34
Урок 3. Основы построения CSS	35
Наследование	35
Контекстные селекторы	37
Классы	38
Теги <i><DIV></i> и <i></i>	42
Абсолютное позиционирование	45
<i>Z-index</i>	47
Каскадирование	48
Зачетный класс	49
Задания	52
Урок 4. Позиционирование, <i>z-index</i>	54
Позиционирование	54
Иерархия кода страницы	54
Абсолютное позиционирование	55
Относительное позиционирование	59
Смешанное позиционирование	61
<i>Z-index</i>	69
Зачетный класс	70
Задания	71
ЧАСТЬ II. ОСНОВЫ JavaScript	75
Урок 5. Первый скрипт	77
Классический HTML болен статичностью	77
JavaScript? Что это такое?	77
Как скрипт «монтируется» в HTML?	78
Зачетный класс	81
Задания	82
Дополнительный материал	82
Урок 6. Переменные, константы, выражения	87
Переменные и константы	87
Описание переменных	88
Выражения	89
Команды	89
Зачетный класс	91
Задания	94
Дополнительный материал	95

Урок 7. Ветвления	105
Ввод информации.....	105
Условная команда.....	106
Переключатель.....	108
Логические операции.....	110
Блок.....	110
Конструируем небольшой экзамен.....	111
Арифметические и строковые выражения как условия.....	113
Значения логического типа.....	114
Трудные условия.....	114
Ввод альтернативы.....	115
Зачетный класс.....	117
Задания.....	120
Дополнительный материал.....	120
Урок 8. Повторения	125
Команды цикла.....	125
Команды <i>break</i> и <i>continue</i>	127
Зачетный класс.....	129
Задания.....	131
Дополнительный материал.....	132
Урок 9. Функции	136
Описание и вызов функции.....	136
Проектирование «сверху вниз».....	138
Оборонительное программирование.....	145
Как браузер разбирается в типах переменных?.....	147
Зачетный класс.....	148
Задания.....	151
Дополнительный материал.....	152
Урок 10. Объект = методы + свойства	159
Понятие объекта.....	159
Доступ к свойствам и методам.....	161
Встроенные объекты и объекты пользователя.....	161
Зачетный класс.....	167
Дополнительный материал.....	169
ЧАСТЬ III. ОБЪЕКТНАЯ МОДЕЛЬ БРАУЗЕРА	193
Урок 11. Объекты и события браузера	195
Объект <i>window</i>	195
Объект <i>document</i>	198

События.....	200
Зачетный класс.....	205
Задания.....	205
Дополнительный материал.....	207
Урок 12. Формы (кнопки, строки ввода).....	212
Принцип программного управления.....	212
Блок <i><Form></i>	213
Элемент <i><INPUT></i>	216
Зачетный класс.....	221
Задания.....	223
Дополнительный материал.....	225
Урок 13. Формы (флажки, радиокнопки).....	231
Флажки.....	231
Радиокнопки.....	237
Зачетный класс.....	241
Задания.....	243
Дополнительный материал.....	245
Урок 14. Формы (меню, многострочное поле).....	248
Меню <i><SELECT></i>	248
Многострочное поле <i><TEXTAREA></i>	254
Зачетный класс.....	259
Задания.....	265
ЧАСТЬ IV. ПЕРВЫЙ ПРАКТИКУМ НА «КУХНЕ» СИДОРОВА.....	267
Урок 15. Конструирование объектов.....	269
Объект для хранения данных.....	269
Построение сложного объекта.....	277
Задания.....	288
Урок 16. Меню.....	289
Меню ссылок на UL.....	290
Меню ссылок на <i><SELECT></i>	292
Меню на стандартных кнопках.....	298
Меню на графических кнопках.....	300
Задания.....	305
Урок 17. Роботландский Плюсик.....	307
Плюсик.....	307
Техническое задание на гипертекстового Плюсика.....	309

Программирование стека	311
Протокол. Откатка. Накатка	316
Плюсик работает!	327
ЧАСТЬ V. CSS + JAVASCRIPT	341
Урок 18. Программное управление стилями	343
Принцип программного управления стилями	343
Доступ к элементам стиля	343
Построение динамического меню	347
Движение на экране	351
Задания	354
Урок 19. Управление содержимым страницы	355
Свойство <i>innerText</i>	355
Свойство <i>outerText</i>	356
Свойство <i>innerHTML</i>	356
Свойство <i>outerHTML</i>	357
«Крестики-нолики»	357
Модель _«плавающего»_ фрейма	363
Задания	366
ЧАСТЬ VI. ВТОРОЙ ПРАКТИКУМ НА «КУХНЕ» СИДОРОВА	369
Урок 20. Второй практикум на «кухне» Сидорова: Конюх	371
Техническое задание	372
Урок 21. Стилиевые решения	376
Урок 22. Скрипты	389
Перетаскивание элементов по экрану	389
Модальные окна	398
Полный код Конюха	405
ЧАСТЬ VII. ПРИЛОЖЕНИЯ	407
Приложение 1. Справочник CSS	409
Единицы измерения	409
Структура справочных таблиц	410

Приложение 2. Справочник JavaScript	424
Зарезервированные слова	424
Конструкции	424
Операции	435
Встроенные функции	444
Встроенные объекты	446
Приложение 3. Справочник. Объекты и события браузера.....	467
<i>window</i>	467
Примеры работы с объектом <i>window</i>	469
<i>document</i>	475
<i>event</i>	482
<i>history</i>	485
<i>location</i>	486
<i>navigator</i>	489
<i>screen</i>	491
События браузера	493
Приложение 4. Коды испытателей.....	495
Испытатель 1. Сравнения (<i>Урок 9</i>).....	495
Испытатель 2. Открытие окна (<i>Урок 11</i>)	496
Испытатель 3. Кнопка с секретом (<i>Урок 11</i>).....	498
Испытатель 4. <i>value</i> и <i>defaultValue</i> (<i>Урок 12</i>).....	500
Испытатель 5. Угадай пароль (<i>Урок 12</i>).....	502
Литература	504

Введение

Что такое DHTML?

Суть построения «живых» гипертекстовых приложений достаточно проста. Браузер как интерпретатор HTML-кода реализует две функции: визуальную и моделирующую. Визуальное действие очень заметно: браузер отображает документ на экране. Но на этом работа не заканчивается. Браузер строит в памяти компьютера программную модель документа, в которой каждому тегу из HTML-файла соответствует отдельный объект.

Посмотрите на схему (рис. 1). Построение объектной модели документа (она носит иерархический характер) браузер начинает с объекта `window`. В этом объекте описаны свойства окна, в котором показан документ. В состав объекта `window` входит объект `document`, который соответствует контейнеру `<HTML>...</HTML>`. Объект `document` включает в себя (среди прочих) объект-массив `images`, который содержит информацию обо всех тегах `` исходного HTML-файла. В частности, первая картинка описывается в этом массиве объектом `images[0]`, вторая — объектом `images[1]` и так далее.

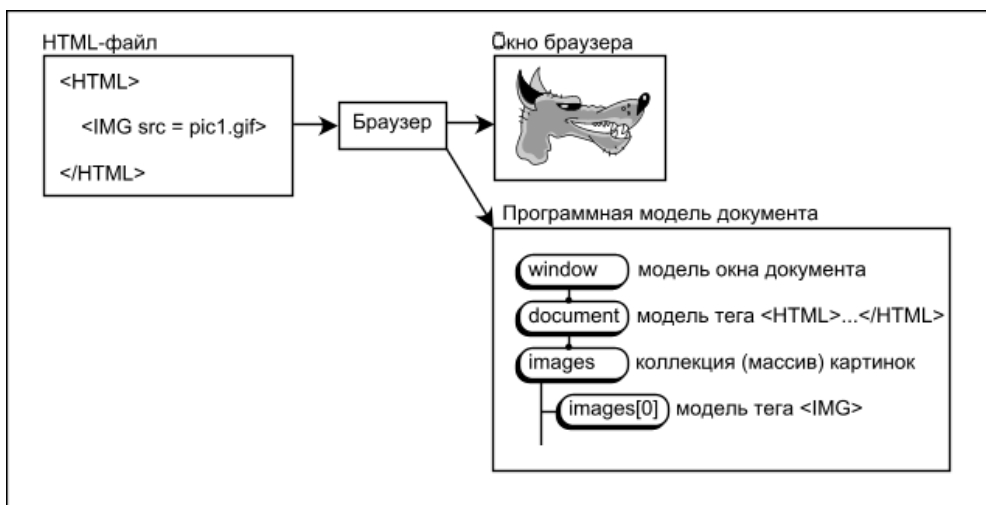


Рис. 1. Браузер строит изображение и объектную модель документа

Документ на экране станет «живым», если в HTML-коде предусмотреть не только описание тегов, но и воздействия на объектную модель при возникновении того или иного события. Можно, например, при наступлении события «курсор мыши над картинкой» изменить в объекте `images[0]` свойство `src` с `pic1.gif` на `pic2.gif` (то есть заменить одну картинку другой). И мы увидим, как волк сердится на компьютерную мышь (рис. 2).

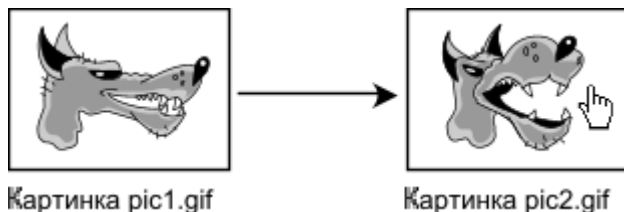


Рис. 2. Замена изображения во время просмотра документа

Программные коды, оживляющие гипертекст (их называют «скриптами», или «сценариями»), записываются на специальных языках программирования. Самым популярным языком, коды которого можно встраивать в HTML-тексты, является JavaScript. Именно этот язык рассмотрен в книге и именно он использован в ней как инструмент создания динамических интерактивных приложений.

Что такое CSS?

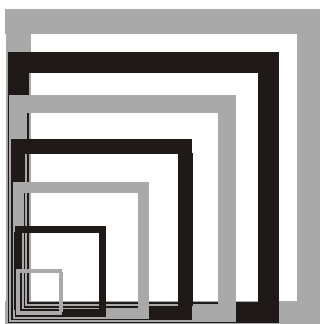
CSS — Cascading Style Sheets (каскадные таблицы стилей) — это средство, позволяющее отделить содержание гипертекстовых страниц от формы их представления на экране, а в паре со сценариями — достичь такой динамичности и интерактивности, при которых гипертекстовое приложение становится практически неотличимо от мультимедийного продукта, созданного в профессиональных программистских оболочках.

Технология CSS допускает, например, абсолютное позиционирование, т. е. позволяет выводить элементы на экран, используя реальные координаты. Программное изменение координат в объектной модели вызывает изменение положения элемента на экране.

Наряду с двумя координатами x и y при помощи CSS можно задавать еще и третью координату — z -индекс. Третья координата определяет номер слоя, в который помещается элемент. Таким образом, при движении в многослойном пространстве одни элементы могут проходить над или под другими.

Но даже при проектировании статичных гипертекстов технология CSS весьма полезна:

- CSS позволяет сократить объем HTML-файла за счет исключения из кода страницы тегов и атрибутов, ответственных за визуальное представление элементов на экране (теги: ``, `<CENTER>`, ``, `<I>`, `<S>`, `<U>` и т. д. ; атрибуты: `align`, `color`, `bgcolor`, `size`, `width`, `height` и т. д.);
- CSS предлагает гораздо больше средств для управления внешним видом элемента, чем обычные теги и атрибуты HTML;
- разработчик получает возможность менять внешний вид сотни гипертекстовых страниц, внося правки только в один файл с описаниями стилевых свойств.



Часть I

ОСНОВЫ CSS

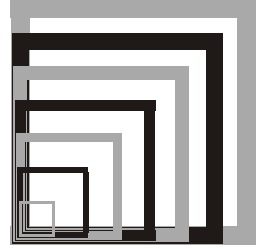
Урок 1. CSS в HTML-коде

Урок 2. Обзор свойств CSS

Урок 3. Основы построения CSS

Урок 4. Позиционирование, *z-index*

Урок 1



CSS в HTML-коде

Стиль для отдельного тега

CSS — Cascading Style Sheets (каскадные таблицы стилей) — это средство, позволяющее задавать различные визуальные свойства HTML-тегам.

Можно указать, например, как должен выглядеть на экране конкретный абзац `<P>`:

```
<P style="font-size:1.5cm; color:green">
```

К этому абзацу применено стилевое определение.

Стиль задается атрибутом `style`. В приведенном примере браузеру дано указание вывести абзац зелеными буквами размером в 1.5 сантиметра. Силевые определения задаются при помощи указаний (рис. 1.1)

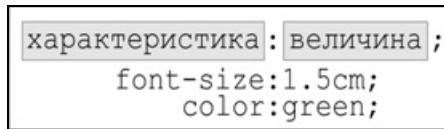


Рис. 1.1. Формат стилевых определений

Указания отделяются друг от друга символом «;».

Атрибут `style` можно использовать практически в каждом теге, задавая его специфические свойства. Посмотрите еще один пример внедрения стилей в теги.

Страница на рис. 1.2 построена таким кодом (листинг 1.1).

Листинг 1.1

```
<HTML>  
<HEAD>  
<META http-equiv="Content-Type"
```

```

        content="text/html; charset=windows-1251">
<TITLE>Стиль для отдельного тега</TITLE>
</HEAD>
<BODY bgcolor=white text=black>
<H2>Стиль для отдельного тега</H2>
<HR>
<UL>
    <LI>Это обычный текст.
    <LI style="color:red;font-size:1cm;font-style:italic">
        Красные наклонные буквы высотой в 1 сантиметр.
    <LI style="margin-left:2cm">
        Отступ слева в 2 сантиметра.
</UL>
</BODY>
</HTML>

```

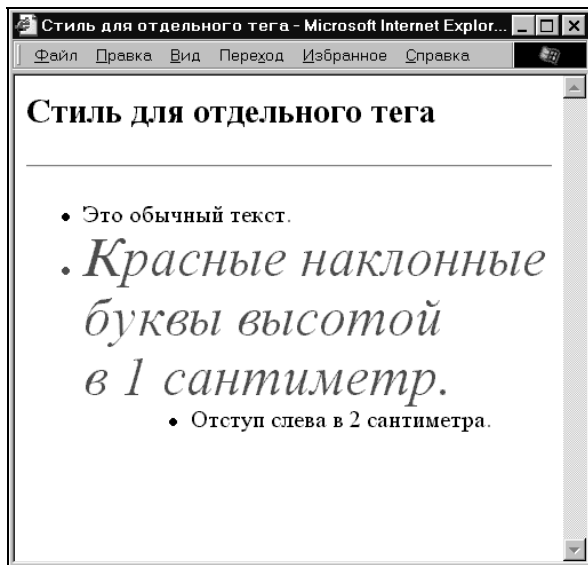


Рис. 1.2. Стиль для одного тега

Стиль для отдельного HTML-файла

Можно задавать стиль для тега или группы тегов так, чтобы определение работало на протяжении всего HTML-документа. Например, можно указать вид всех заголовков. Для этого достаточно написать определение (листинг 1.2) в головной части документа.

Листинг 1.2

```
<HTML>
<HEAD>
  <META http-equiv="Content-Type"
    content="text/html; charset=windows-1251">
  <TITLE>Стиль для отдельного файла</TITLE>
  <STYLE type="text/css">
    <!--
      H1, H2, H3, H4, H5, H6
      {
        text-align:right;
        color:maroon;
        font-family: "Arial Cyr", Geneva, Helvetica, sans-serif;
      }
    -->
  </STYLE>
</HEAD>
<BODY bgcolor=#DFFD05 text=black>
  <H2>Стиль для отдельного файла</H2>
  <HR>
  <P>Это обычный текст
  <H3>Это заголовок</H3>
  <P>Это снова обычный текст
</BODY>
</HTML>
```

Браузер отображает заголовки в этом документе рубленным шрифтом цвета maroon и выравнивает их по правому краю экрана (рис. 1.3).

Такое поведение браузера соответствует инструкции (листинг 1.3).

Листинг 1.3

```
<STYLE type="text/css">
  <!--
    H1, H2, H3, H4, H5, H6
    {
      text-align:right;
      color:maroon;
      font-family: "Arial Cyr", Geneva, Helvetica, sans-serif;
    }
  -->
```



```
-->
</STYLE>
```

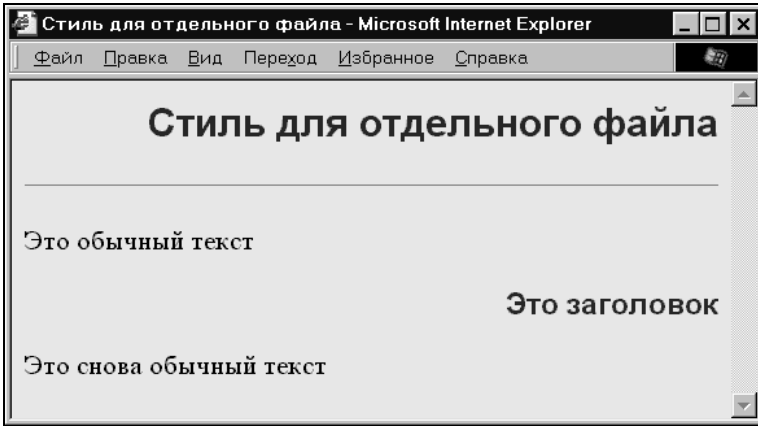


Рис. 1.3. Стиль для отдельного файла

Стилевые определения (*селекторы*) располагаются внутри блока `<STYLE>...</STYLE>` и «запаковываются» в HTML-комментарий (для браузеров, которые не поддерживают CSS).

Стилевое определение имеет вид:

Имя тега (или имена тегов через запятые)

```
{
  характеристика: величина;
  ...
  характеристика: величина;
}
```

В приведенном выше примере использованы три характеристики:

<code>text-align:right;</code>	—	задает выравнивание по левому краю;
<code>color:maroon;</code>	—	задает цвет maroon;
<code>font-family:"Arial Cyr", Geneva, Helvetica,sans-serif;</code>	—	задает рубленый шрифт.

Заголовки будут выводиться шрифтом Arial Cyr, если, конечно, этот шрифт есть в компьютере пользователя. Если шрифта нет, браузер последовательно ищет шрифты Geneva, Helvetica или, в конце концов, какой-нибудь рубленый шрифт (указание «sans-serif»). В случае полной неудачи браузер выведет текст «шрифтом по умолчанию», т. е., как правило, шрифтом Times New Roman.

Стиль для нескольких HTML-файлов

Обычной практикой является указание стилей в отдельном файле. Для таких файлов используют расширение `css`. Например, можно в файле `style.css` записать:

```
BODY {margin-left:40px;}
H1,H2,H3,H4,H5,H6
{
    text-align:right;
    color:maroon;
    font-family:"Arial Cyr",Geneva,Helvetica,sans-serif;
}
```

Для подключения этих указаний в разделе `<HEAD>...</HEAD>` HTML-файла нужно поместить ссылку:

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

Такой способ расположения стилевых определений очень удобен. На один и тот же стилиевой файл могут ссылаться многие HTML-документы. Изменения в этом единственном файле скажутся на внешнем виде десятка (а то и сотни) страниц.

Обратите внимание на стилевое указание `BODY {margin-left:40px;}`.

Оно задает экранный отступ слева в 40 пикселей для всего документа в целом. В силу такого определения элементы не будут «наползать» на вертикальную фоновую полосу, моделирующую левое страничное поле. Как видите, для задания страничного отступа можно обойтись без таблиц и «распорок», о которых говорилось в книге «Web-конструирование. HTML».

Комбинирование стилей

Были показаны три способа внедрения стилей в HTML-коды:

- указания в отдельном теге;
- указания в заголовке HTML-файла;
- указания в отдельном CSS-файле.

А что если комбинировать эти способы? Какой из них окажется самым «сильным» для конкретного тега?

Проделаем серию опытов.

Документ без CSS-указаний

В примере, код которого приводится ниже (листинг 1.4, рис. 1,4), заголовки выводятся черным цветом по белому.

Листинг 1.4

```
<HTML>
  <HEAD>
    <TITLE>Пример 1</TITLE>
  </HEAD>
  <BODY bgcolor=white text=black>
    <H1>Заголовок 1</H1>
    <H2>Заголовок 2</H2>
    <H3>Заголовок 3</H3>
  </BODY>
</HTML>
```

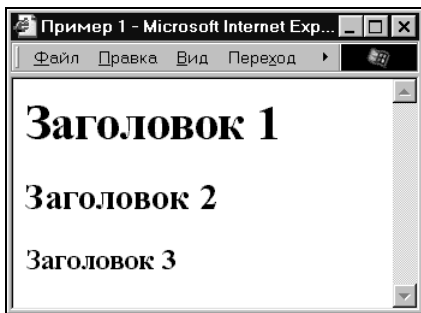


Рис. 1.4. Документ без CSS-указаний

CSS-указания в отдельном теге

Этот код (листинг 1.5) отобразит первые два заголовка черным цветом, а последний — красным.

Листинг 1.5

```
<HTML>
  <HEAD>
    <TITLE>Пример 2</TITLE>
  </HEAD>
  <BODY bgcolor=white text=black>
    <H1>Заголовок 1</H1>
```

```
<H2>Заголовок 2</H2>
<H3 style="color:red">Заголовок 3</H3>
</BODY>
</HTML>
```

CSS-указания в <HEAD>

Здесь (листинг 1.6) первые два заголовка выводятся синим цветом, а последний — красным.

Листинг 1.6

```
<HTML>
<HEAD>
  <STYLE type="text/css">
    <!--
      H1,H2,H3
      {
        color:blue;
      }
    -->
  </STYLE>
  <TITLE>Пример 3</TITLE>
</HEAD>
<BODY bgcolor=white text=black>
  <H1>Заголовок 1</H1>
  <H2>Заголовок 2</H2>
  <H3 style="color:red">Заголовок 3</H3>
</BODY>
</HTML>
```

CSS-указания в CSS-файле

Файл prim.css содержит:

```
H1,H2,H3
{
  color:green;
}
```