**Назначение CSS**

Дизайн Web-узлов — это точное размещение компонентов HTML-страниц относительно друг друга в рабочей области окна браузера.

Недостатки такого определения Web-дизайна очевидны. В нем не учтены ни цвет, ни форма, ни другие свойства компонентов HTML-страниц. Главное в этом определении — показать ограниченность возможностей HTML-разметки. Позиционирование компонентов на странице является одним из самых слабых мест в HTML.

К компонентам страницы относятся: блоки текста, графика и встроенные приложения. Размер и границы каждого из этих компонентов в рамках HTML-разметки задаются с разной степенью точности. Размер графики и приложений можно задать с точностью до пикселя. Размеры текстовых блоков в HTML задать нельзя: они вычисляются браузером исходя из относительного размера шрифта по умолчанию.

Автор страницы не может заранее определить настройки браузера пользователя, что существенно ограничивает число вариантов представления информации на странице.

Нельзя сказать, что разработчики браузеров не пытались изменить данную ситуацию. В ранних версиях браузеров CERN для платформы NEXT и в браузерах WWWC автор страницы имел возможность переопределять настройки браузера по умолчанию через HTML-разметку. Но этот подход не получил продолжения в коммерческих продуктах.

Другой способ управления настройками браузера — программирование на JavaScript. Бурное развитие этого языка позволяет говорить о возможности полного контроля над процессом отображения HTML-страниц. Недостаток JavaScript — отказ от декларативного характера разметки и относительно большой объем кода для переопределения свойств элементов разметки.

Спецификация CSS (Cascading Style Sheets) позволяет остаться в рамках декларативного характера разметки страницы и полностью контролировать форму представления элементов HTML-разметки.

Каскадные таблицы стилей призваны разрешить противоречие между точностью определения размеров картинок и приложений, с одной стороны, и точностью определения размеров блоков текста и его начертания — с другой.

Таблицы стилей также позволяют определить цвет и начертание текстового фрагмента, изменять эти параметры внутри текстового блока, выполнять выравнивание текстового блока относительно других блоков и компонентов страницы.

Наличие подобных возможностей позволяет говорить о CSS как о средстве разделения логической структуры документа и формы его представления. Логическая структура документа определяется элементами HTML-разметки, в то время как форма представления каждого из этих элементов задается CSS-описателем элемента.

CSS позволяет полностью переопределить форму представления элемента разметки по умолчанию. Например, <I>...</I> определяет отображение текста курсивом:

<I>Отобразим текст курсивом</I>

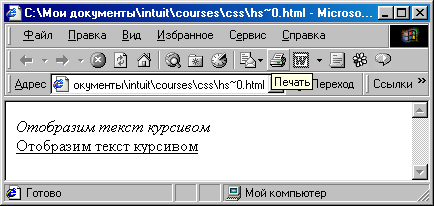
А теперь переопределим стиль отображения для элемента разметки I ([открыть](https://www.intuit.ru/EDI/11_04_18_2/1523398877-17984/tutorial/101/objects/1/files/example1.htm)):

<I STYLE="text-decoration:underline;

font-style:normal;">

Отобразим текст курсивом

</I>



**Рис. 1.1.**

Этот пример показывает, что привычный стиль отображения элементов может быть полностью изменен при помощи CSS. В данной технологии HTML-разметка носит чисто декларативный характер.

Практическое значение CSS для Web-инжиниринга (совокупности технологий разработки и сопровождения Web-узлов) заключается в том, что процесс создания узла можно формализовать и представить в виде последовательности действий:

* необходимо определиться с номенклатурой страниц, т.е. все страницы проектируемого Web-узла разбить на типы, например домашняя страница, навигационные страницы, информационные страницы, коммуникационные страницы и т.п. У каждого узла этот перечень может быть свой;
* для каждого типа страниц требуется разработать определенную логическую структуру (стандартный набор компонентов страницы);
* следует создать навигационную карту узла и форму ее реализации на страницах;
* для каждого стандартного компонента страницы нужно разработать стиль его отображения (CSS-описатель);
* теперь остается только рисовать картинки, создавать анимацию, писать программы, вручную вводить текст и графику или генерировать содержимое страниц автоматически при обращении к ним.

Объяснив таким образом роль и назначение CSS среди многообразия Web-технологий, мы переходим непосредственно к обсуждению применения каскадных таблиц стилей.

### Способы применения CSS

Под способами применения CSS мы в данном разделе понимаем форму декларирования стиля на HTML-странице и форму связывания описания стиля отображения элемента разметки с самим элементом. Речь идет о том, где и в какой форме автор страницы (или дизайнер) описывает стиль, и как и в какой форме на него ссылается.

Итак, различают четыре способа применения стилей:

* переопределение стиля в элементе разметки ;
* размещение описания стиля в заголовке документа в элементе STYLE ;
* размещение ссылки на внешнее описание через элемент LINK ;
* импорт описания стиля в документ.

#### Переопределение стиля

<H1 STYLE="font-weight:normal;

font-style:italic;

font-size:10pt;">

Заголовок первого уровня

</H1>

Атрибут style можно применить внутри любого элемента разметки. Например, мы можем через style определить ширину и выравнивание элемента hr (горизонтальное отчеркивание):

<HR STYLE="width:100px;">

Очевидно, что не все параметры стиля можно установить для конкретного элемента разметки. О типах элементов и соответствующих параметрах стилей мы поговорим в разделе "Понятия блочного и строкового элементов".

Здесь же нужно отметить следующее: стили разработаны в первую очередь для управления отображением текста. Не следует увлекаться стилями при управлении отображением нетекстовых элементов HTML-разметки.

#### Элемент STYLE

Применение элемента STYLE — это основной способ внедрения каскадных таблиц стилей в ткань HTML-документа. Помимо управления отображением элементов разметки, элемент STYLE позволяет описывать стилевые свойства элементов, которые можно изменять при программировании на JavaScript.

Элемент STYLE дает возможность определить стиль отображения для:

* стандартных элементов HTML-разметки ;
* произвольных классов ( селектор   CLASS );
* НTML-объектов ( селектор   ID ).

К сожалению, работа с селекторами в браузерах различных производителей может преподносить различного рода сюрпризы, для избежания которых следует придерживаться стандартов W3C.

Понятие селектора, применение селекторов и формальный синтаксис CSS мы обсудим в разделах "Синтаксис" и " Наследование и переопределение ".

Стандартные элементы разметки описываются в элементе STYLE следующим образом:

<HEAD>

<STYLE>

p { color:darkred;text-align:justify;

font-size:8pt; }</STYLE>

</HEAD>

<BODY>

...

<P>

Этот параграф мы используем как пример

применения описания стиля для стандартного

элемента HTML-разметки.

</P>

...

</BODY>

Теперь все параграфы документа будут отображаться стилем из элемента STYLE, если только стиль не будет каким-либо способом переопределен. В STYLE можно определить стиль любого элемента разметки.

#### Ссылка на внешнее описание

Ссылка на описание стиля, расположенного за пределами документа, осуществляется при помощи элемента LINK, который размещают в элементе HEAD . Внешнее описание может представлять собой файл, содержащий описание стилей. Описание стилей в этом файле будет по синтаксису в точности совпадать с содержанием элемента STYLE.

Ниже приведен пример ссылки на внешнее описание стилей:

<LINK TYPE="text/css" REL="stylesheet"

HREF="http://intuit.ru/my\_css.css">

Здесь важны значения атрибутов REL и TYPE. Атрибут REL должен иметь значение stylesheet. Type может принимать значения text/css или text/javascript. Второй тип описания стилей введен Netscape. Его мы в данном учебном курсе не обсуждаем.

Атрибут HREF задает универсальный локатор ресурса (URL) для внешнего файла описания стилей. Это может быть ссылка на файл с любым именем, а не только на файл с расширением \*.css.

#### Импорт описания стилей

Импорт описателей стилей в некотором смысле составляет конкуренцию представленному выше указанию на внешний описатель стиля.

Импортировать стиль можно либо внутрь элемента STYLE, либо внутрь внешнего файла, который представляет собой описатель стиля. Оператор импорта стиля должен предшествовать всем прочим описателям стилей:

<STYLE>

@import url(http://intuit.ru/style.css)

A { color:cyan;text-decoration:underline; }

</STYLE>

Импортируемый стиль можно переопределить либо через описатель элемента в STYLE, либо через атрибут элемента STYLE.

### Синтаксис

Формально стиль отображения элементов разметки задается ссылкой в элементе разметки на селектор стиля. Синтаксис описания стилей в общем виде представляется следующим образом:

selector[, selector[, ...]]

{ attribute:value;

[attribute:value;...] }

или

selector selector [selector ...]

{ attribute:value;

[attribute:value;...] }

В первом варианте перечислены селекторы, для которых действует данное описание стиля. Второй вариант задает иерархию вложенности селекторов, для совокупности которых определен стиль. Напомним, что речь в данном случае идет об описаниях стилей в нотации text/css. Описания стилей размещаются либо внутри элемента STYLE, либо во внешнем файле.

В качестве селектора можно использовать имя элемента разметки, имя класса и идентификатор объекта на HTML-странице.

Атрибут (attribute) определяет свойство отображаемого элемента, например левый отступ параграфа (margin-left), а значение (value) — значение этого атрибута, например, 10 типографских пунктов (10 pt).

#### Селектор — имя элемента разметки

Когда автор Web-узла хочет определить общий стиль всех страниц, он просто прописывает стили для всех элементов HTML-разметки, которые будут использоваться на страницах. Это дает возможность скомпоновать страницы из логических элементов, а стиль отображения элементов описать во внешнем файле.

Такой способ создания сайта позволяет автору изменять внешний вид всех страниц путем внесения изменений в файл описания стилей, а не в файлы HTML-страниц.

Внешний файл при этом может выглядеть следующим образом:

I, EM {color:#003366;font-style:normal;}

A I {font-style:normal;font-weight:bold;

text-decoration:line-through;}

В первой строке этого описания перечислены селекторы-элементы, которые будут отображаться одинаково:

<I>Это курсив</I> и это тоже <EM>курсив</EM>

Последняя строка определяет стиль отображения вложенного в гипертекстовую ссылку курсива:

<A NAME=empty><I>intuit</I></A>

В данном случае переопределение состоит в том, что текст отображается внутри гипертекстовой ссылки перечеркнутым, причем жирным шрифтом.

#### Селектор — имя класса

Имя класса не является каким-либо стандартным именем элемента HTML-разметки. Оно определяет описание класса элементов разметки, которые будут отображаться одинаково. Для того, чтобы отнести элемент разметки к тому или иному классу, нужно воспользоваться его атрибутом CLASS ([открыть](https://www.intuit.ru/EDI/11_04_18_2/1523398877-17984/tutorial/101/objects/1/files/example2.htm)):

<STYLE>

.test {color:white;background-color:black;}

</STYLE>

...

<P CLASS="test">

Этот параграф мы отобразим белым цветом по

черному фону

</P>

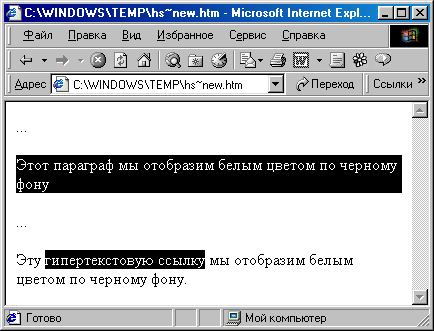
...

<P>

Эту <A CLASS="test">гипертекстовую ссылку</A>

мы отобразим белым цветом по черному фону.

</P>



**Рис. 1.2.**

Таким образом в любом элементе разметки можно сослаться на описание класса отображения. При этом совершенно необязательно, чтобы элементы разметки были однотипными. В примере к одному классу отнесены и параграф, и гипертекстовая ссылка в другом параграфе.

Лидирующую точку в имени класса можно опустить. Она задается из соображений сохранения единства описания. Например, можно определить классы отображения однотипных элементов разметки:

a.menu { color:red;background-color:white;

text-decoration:none; }

a.paragraph { color:navy;

text-decoration:underline; }

В данном примере класс гипертекстовых ссылок menu имеет одно описание стиля, а класс гипертекстовых ссылок paragraph — совершенно другое. При этом каждый из этих классов нельзя применить к другим элементам разметки, например, параграфу или списку. Если имя элемента разметки не задано, это означает, что класс можно отнести к любому элементу разметки — корневой класс описания стилей. Это очень похоже на обозначение имени корневого домена в системе доменных имен. Собственно ничего удивительного здесь нет, т.к. система классов объектов на HTML-странице представляет собой дерево. Элементы разметки — это узлы дерева.

#### Селектор — идентификатор объекта

Объектная модель документа (Document Object Model) описывает документ как дерево объектов. Объектами являются: сам документ, его разделы (элемент DIV ), картинки, параграфы, приложения и т.п. Каждому из объектов можно дать имя и обращаться к нему по имени. Данная возможность используется при программировании страниц на стороне клиента.

Применение идентификатора объекта оправдано еще и в случае модификации атрибута описания стиля для данного объекта в его CSS-описании. Вместо двух описаний классов, которые отличаются только одним из параметров, можно создать одно описание класса и описание идентификатора объекта. Описание стиля для объекта задается строкой, в которой селектор представляет собой имя этого объекта с лидирующим символом "#":

a.mainlink { color:darkred;

text-decoration:underline;

font-style:italic; }

#blue { color:#003366 }

...

<A CLASS=mainlink>основная гипертекстовая

ссылка</A>

<A CLASS=mainlink ID=blue>модифицированная

гипертекстовая ссылка</A>

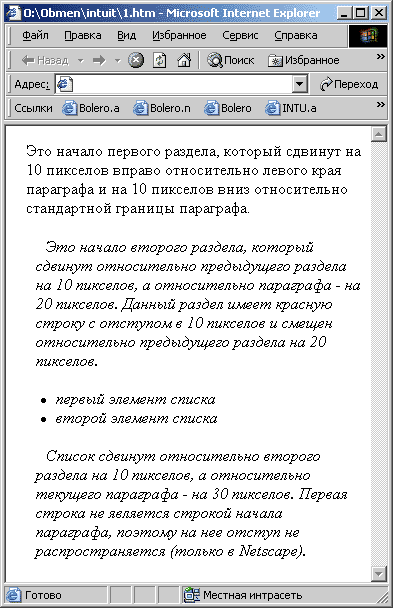
Следует отметить, что интерпретация идентификаторов объектов в Internet Explorer и Netscape Navigator различна. Существует еще атрибут name у элемента разметки. При идентификации объекта Netscape Navigator обычно имеет дело именно с этим атрибутом, а Internet Explorer — с атрибутом ID.

Различия в интерпретации ID в браузерах при декларативном использовании CSS не очень страшны. Другое дело, если автор решится программировать стили, т.е. изменять значения атрибутов описателей стилей. В этом случае разница объектных моделей документов в Netscape Navigator и Internet Explorer проявится в полной мере. Фактически, придется для каждого из браузеров разрабатывать совершенно разные страницы.

**Наследование и переопределение**

При обсуждении технических спецификаций часто бывает полезно вникнуть в смысл названия. В названии принято точно определять суть и назначение стандарта или спецификации. Описание стилей отображения элементов HTML-разметки носит название "Каскадные таблицы стилей". Со словом "стилей" все более-менее понятно. Под словом "таблицы" следует понимать набор свойств элемента разметки, который можно представить в виде строки в таблице свойств, т.е. элементы разметки — строки, а свойства — столбцы. А вот слово "каскадные" требует пояснения.

Во-первых, существует иерархия элементов разметки (дерево объектов на странице). Во-вторых, свойства этих объектов могут наследоваться. Таким образом в дереве объектов образуется ветвь, которая ведет к листу дерева — элементу разметки, например, элементу списка или параграфу. Его свойства определяются элементами разметки, в которые вложен элемент, и описателями стиля для данного элемента:



Предыдущий текст закодирован в терминах разделов и списка следующим образом:

<DIV STYLE="margin-left:10px;margin-top:10px;">

Это начало первого раздела, который сдвинут

на 10 пикселов вправо относительно левого

края параграфа и на 10 пикселов вниз

относительно стандартной границы параграфа.

<DIV STYLE="margin-left:10px;margin-top:20px;

text-indent:10px;font-style:italic;">

Это начало второго раздела, который

сдвинут относительно предыдущего раздела

на 10 пикселов, а относительно параграфа —

на 20 пикселов. Данный раздел имеет красную

строку с отступом в 10 пикселов и смещен

относительно предыдущего раздела на 20

пикселов.

<UL STYLE="margin-left:10px;">

<LI>первый элемент списка

<LI>второй элемент списка

</UL>

Список сдвинут относительно второго раздела

на 10 пикселов, а относительно текущего

параграфа — на 30 пикселов. Первая строка не

является строкой начала параграфа, поэтому на

нее отступ не распространяется (только в

Netscape).

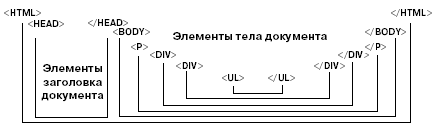
</DIV>

</DIV>

Таким образом отступы отсчитываются относительно элемента, в который вложен текущий элемент. Все параметры, которые не были переопределены в текущем элементе, наследуются из старшего по иерархии элемента. Последнее хорошо продемонстрировано в применении стилей отображения списка, который вложен в раздел и поэтому отображается курсивом.

Когда объяснение некоторого феномена HTML-разметки растягивается на несколько параграфов, имеет смысл воспользоваться приведенной ниже графической схемой построения страницы.

При использовании стилей действуют следующие правила старшинства стилей:



**Рис. 1.3.**

* сначала применяются стили браузера по умолчанию;
* стили браузера по умолчанию переопределяются прилинкованными стилями (элемент LINK заголовка документа);
* прилинкованные стили переопределяются описаниями стилей в элементе STYLE ;
* стили элемента STYLE переопределяются атрибутом STYLE в любом из элементов разметки.

Не все атрибуты стиля могут наследоваться. Например, "набивка" (отступ содержания элемента от его границ) элемента BODY не наследуется вложенными в него элементами и определяется по умолчанию или прописывается для каждого элемента отдельно. Алгоритмы наследования в Internet Explorer и в Netscape Navigator разные, поэтому для единства отображения элементов следует прописывать стиль по максимуму атрибутов.