Использование каскадных таблиц стилей (CSS)

- **Общие положения**
  - Каскадные таблицы стилей или CSS (от английского Cascading Style Sheets) позволяют разделить смысловое содержимое странички и его оформление.
  - Таблицы стилей, слои и универсальная объектная модель браузера были введены в HTML 4.0 (Dynamic HTML).

Использование каскадных таблиц стилей (CSS)

- **Общие положения**
  - Четвертая версия, рекомендует создавать странички таким образом, чтобы они могли быть воспроизведены на любом устройстве - будь это 21" монитор или маленький экран сотового телефона.
  - Для этого все оформление рекомендуется вынести во внешний стилевой файл. Основная же страничка будет содержать только информацию и ссылки на необходимые стили.

Использование каскадных таблиц стилей (CSS)

- **Общие положения**
  - При показе странички конкретному устройству должна быть задействована соответствующая случаю таблица стилей.
  - Таблицу стилей нужно написать всего один раз при создании сайта для каждого из устройств, на котором планируется вывод информации.
  - Таблица стилей может быть единой для целого сайта. Тогда не нужно будет повторять одни и те же описания стилей на каждой из страниц.

Использование каскадных таблиц стилей (CSS)

- **Общие положения**
  - Размещение всей стилевой информации в одном внешнем файле открывает и другие полезные возможности - ведь изменив содержимое только одного (!) стилевого файла, мы можем в считанные секунды сменить весь дизайн сайта. Причем никаких других переделок не понадобится.

Подключение таблиц стилей

- Для осуществления этой задачи мы можем воспользоваться одним из 3-х предлагаемых методов:
  - внешний файл
  - описание в секции заголовка
  - inline-описание
- Таблицы стилей, подключенные этими способами, называют, соответственно, связанными, внедренными или встроенными.

inline-описание

- Самое простое, так называемое inline-описание, встроено в тег:
- **<P style="color:red; text-align:center;">Этот текст переопределен стилем</P>**
- При помощи дополнительного атрибута style мы можем определить нужные нам стилевые параметры в любом теге. Это самый легкий способ, и действует он в пределах лишь одного тега. Этот способ не слишком отличается, к примеру, от прямого описания внешнего вида при помощи тега <font>.

описание в секции заголовка

- Гораздо удобнее заранее определить все нужные стили оформления и впоследствии просто применять их к соответствующим тегам. Это будет второй способ - описание в секции заголовка. Его действие распространяется на всю страничку. Определение стилей происходит при помощи классов, которые представляют собой списки с определением всех необходимых параметров оформления.

описание в секции заголовка

- При использовании этого метода описание стилей необходимо разместить в секции заголовка:
- **<HEAD>....**
- **<STYLE TYPE="TEXT/CSS">**
- **<!—**
- **.HEADER { TEXT-ALIGN : CENTER; FONT-SIZE : 27PT;}**
- **.RED { COLOR : RED; }**
- **-->**
- **</STYLE>**
- **</HEAD>**

описание в секции заголовка

- Теперь эти стили можно применять в любом месте html-кода. Для этого используется следующая конструкция:
- **<P Class=Header>**Этот текст написан стилем header**</P>**
- **<P Class=Red>**Этот текст написан красным цветом**</P>**

описание в секции заголовка

- Кроме определения новых классов мы также имеем возможность переопределять стандартные теги. Например, тег <p>:
- **<Style type="text/css">**
- **<!—**
- **P { text-align : center; font-size : 12pt;}**
- **-->**
- **</Style>**
- Это позволяет легко адаптировать уже существующие странички к использованию стилей. Кроме того, это несколько уменьшает объем файла за счет отсутствия лишних атрибутов *class*.

описание во внешнем файле

- Диапазон его воздействия простирается на все файлы, в которые включено описание. Это способ наиболее соответствует концепции HTML 4.0. В случае, если нам потребуется изменить внешний вид сайта, то будет достаточно скорректировать лишь один этот файл.

описание во внешнем файле

- Создается стилевой файл с описанием всех нужных классов (mystyle.css):
- **.Header { text-align : center; font-size : 27pt;}**
- **.red { color : red; }**
- **P { text-align : center; font-size : 12pt;}**
- А потом ссылка на него внедряется в документ при помощи тега <link>:
- **<Head>....**
- **<Link rel="stylesheet" type="text/css" href="css/mystyle.css" title="mystylesheet">**
- **....**
- **</head>**

Каскадность стилей

- Повторим способы внедрения стилей на страничку:
  - использование отдельного стилевого файла и вставка его при помощи тега <link>
  - описание стиля в заголовке документа
  - применения стиля как параметра в теге.
- Каскадность заключается в том, что стили могут переопределяться. Список способов внедрения стилей соответствует порядку переопределения. Нижерасположенный способ может переопределять вышерасположенный.

Каскадность стилей

- Покажем каскадность на конкретном примере. Предположим, что для всех ссылок в заголовке на нашей страничке определен следующий стиль:
- **<Style type="text/css">**
- **<!—**
- **A { text-decoration: none; color:red; }**
- **-->**
- **</Style>**
- **</Head>**

Каскадность стилей

- Любой текст, который является гиперссылкой, автоматически становится красным и перестает быть подчеркнутым. В конце странички надо указать копирайт, но сделать это не сильно заметно. Однако копирайт также должен быть ссылкой. Сделать это нужно всего лишь в одном месте странички и определять для этого дополнительный класс нецелесообразно.
- В этом случае нам на помощь придет каскадность стилей. Надо локально переопределить цвет ссылки:
- **Copyright © 1998-2001 Gun **

Каскадность стилей

- Сделано это при помощи параметра style, а он действует лишь в пределах того тега, в котором был определен.
- В вышеприведенном примере введен в обращение новый тег **.** Он предназначен специально для таких случаев. Все, что он делает - это определяет некую область, к которой можно применить стиль. Это очень удобный тег, т.к. не вставляет ни до, ни после себя ненужных отбивок (т.е. пустое вертикальное пространство), как это делает тег <p>.

Каскадность стилей

- В каких тегах лучше определять стили посредством класса? Чаще всего для этого используется одна из следующих конструкций:
- ** Что-то **
- **<td class="big">Что-то</td>**
- ** Что-то **
- **Что-то**
- Тег подобен , но только с тем отличием, что делает до и после себя отбивку (точно так же, как и).

Каскадность стилей

- Стиль текста, которым набрано основное содержимое странички, лучше всего сделать переопределением тега <p>, а не определением отдельного класса.
- И небольшое дополнение, связанное с корректным показом в разных браузерах - если Вы используете табличную верстку для дизайна сайта, то определять стиль основного текста нужно не только в теге <p>, но и в <td>, т.к. Netscape категорически отказывается наследовать стили, присвоенные до таблицы.

Синтаксис CSS

- Описание каждого класса делается при помощи конструкции, подобной этой:
- **.small { font-size: 9pt; }**
- Сначала указывается имя класса - оно может быть произвольным, но желательно все-таки давать осмысленное название. Далее, в фигурных скобках {} перечисляются все необходимые параметры для данного класса. Параметры отделяются друг от друга точкой с запятой. Вот еще один пример, в котором используется более длинное описание:
- **.small { font-size: 9pt; color: #eeeeee; text-align: center; }**

Синтаксис CSS

- Заметьте, что в обеих конструкциях имя класса начинается с точки и таким образом определен универсальный класс, т.е. такой, который может быть применен к любому тегу. И делается это при помощи следующей конструкции:
- ** Накладываем стиль на этот текст **
- **<td class=small>Накладываем стиль на этот текст</td>**

Синтаксис CSS

- Еще бывают так называемые теговые классы:
- **p.small { font-size: 9pt; }**
- Класс, определенный таким образом, сработает только в том теге, для которого он предназначен, а для всех остальных будет проигнорирован:
- ** Этот текст будет выведен стилем small **
- **<td class=small>А этот останется неизменным</td>**

Синтаксис CSS

- Мы можем определять параметры не только для одного тега, но и сразу для нескольких. Для этого в определении стиля достаточно перечислить их через запятую:
- **p, td { font-size: 9pt; color:green;}**
- Такой прием называется группировкой, и в данном случае мы определили и для <p>, и для <td> одинаковый размер и цвет текста.
- В случае переопределения существующих тегов, в описании стиля можно указывать не все параметры, а лишь те из них, которые мы хотим изменить. Все остальные параметры примут значения по умолчанию, которые для разных тегов различны.

Псевдоклассы

- В CSS есть такое понятие как псевдокласс. В отличие от обычного класса, действие псевдокласса распространяется не на весь текст, к которому применен данный стиль, а лишь на его часть и возможно лишь в определенном состоянии. Чтобы было понятнее, давайте разберем эффект, при котором ссылки подчеркиваются лишь при наведении на них курсора. Эффект достаточно распространенный. Вот фрагмент таблицы стилей, который позволяет достигать вышеописанного эффекта:
- **a { text-decoration: none; }**
- **a:hover { text-decoration: underline; }**

Псевдоклассы

- **a { text-decoration: none; }**
- **a:hover { text-decoration: underline; }**
- Верхняя строчка - это переопределение стандартного тега <a>, которое запрещает подчеркивать ссылки, а вот нижняя - это определение стиля для псевдокласса *hover*, который описывает стиль ссылки в момент, когда курсор находится над ней.

Псевдоклассы

- А вот и другой пример псевдокласса - определение буквицы вначале абзаца:
- **p:first-letter { font-size: 200%; font-weight: bold; }**
- Заметьте, что и в том, и в другом случае действие стиля распространяется либо на определенное состояние (пользователь собирается щелкнуть по ссылке), либо на фрагмент текста (изменяется только первая буква абзаца). В этом и заключается смысл псевдоклассов.

Комментарии

- Как и в любом достаточно сложном языке, при создании таблицы стилей можно пользоваться комментариями. Их формат аналогичен классическому C:
- /* Этот текст является комментарием */
- Для небольших сайтов эта возможность Вам вряд ли пригодится, а вот при создании сложных, многоуровневых таблиц стилей комментарии могут пригодиться. Кстати, здесь будет уместно привести золотое правило - чем понятнее названа переменная (в данном случае имя класса), тем меньше комментариев необходимо.

Основные параметры CSS

- Все параметры, используемые для определении стиля, условно можно разделить на несколько больших групп:
  - управляющие видом шрифта (гарнитура, кегль, цвет, наклон, жирность,..)
  - управляющие форматированием абзаца (выравнивание, интерлиньяж, расстояние между словами, отступ красной строки,..)
  - управляющие свойствами блока (отступы слева-сверху-справа-снизу, местоположение блока на страничке, видимость блока,..)
  - другие, которые не вписываются ни в одну из перечисленных выше групп (цвет ссылок странички, изменение внешнего вида курсора,..)

Основные параметры CSS

- *Основные параметры шрифта*
- **font-weight: [bold | normal | number] - жирность шрифта**
- **font-style: [normal | italic | oblique] - наклон шрифта**
- **font-size: number - размер шрифта**
- **font-family: name - гарнитура шрифта**
- **color: number - цвет шрифта**
- **background-color: number - цвет подложки**
- **background: url - текстурная подложка**

Основные параметры CSS

- *Основные параметры абзаца*
- text-align: [left|right|center|justify] – выравнивание
- text-indent: number - отступ красной строки
- line-height: number – интерлиньяж
- letter-spacing: number – трекинг
- padding-left: number - отступ от текста слева
- padding-right: number - отступ от текста справа
- padding-top: number - отступ от текста сверху
- padding-bottom: number - отступ от текста снизу
- margin-left: number - отступ от границы слева
- margin-right: number - отступ от границы справа
- margin-top: number - отступ от границы сверху
- margin-bottom: number - отступ от границы снизу

Основные параметры CSS

- ***Единицы измерения в CSS***
- В свойствах, которым требуется указание размеров, можно использовать несколько способов для их задания:
- относительный размер в процентах (%);
- относительный размер при помощи словесного описания **(larger, smaller, xx-small, x-small, small, medium, large, x-large, xx-large);**
- абсолютный размер в типографских единицах - размер может задаваться в пунктах (pt), пиках (pc), пикселях (px), средней шириной буквы "m" (em), средней шириной буквы "x" (ex);

Основные параметры CSS

- ***Единицы измерения в CSS***
- абсолютный размер в стандартных единицах длины - размер может задаваться в сантиметрах (cm), миллиметрах (mm), дюймах (in);
- абсолютный в пикселях (px).
- ***Задание цвета в CSS***
- Цвет для тех свойств, где это нужно, может быть определен одним из трех способов:
- при помощи названия цвета: yellow, red, green, grey,..
- шестнадцатеричным заданием цвета в формате #RRGGBB: #ff0000, #883490, #ffffff,..
- десятичным заданием составляющих цвета в формате rgb(red, green, blue): rgb(255,0,0), rgb(100,23,78),..

примеры описания **CSS**

- Несколько примеров описания таблицы стилей:
- **.epigraph {**
- **font-size: 12pt;**
- **font-style: italic;**
- **text-align: right;**
- **color: rgb(127,127,0);**
- **}**
- **p.big {**
- **font-size: 16px;**
- **font-weight: bold;**
- **color: #ff0000;**
- **}**

примеры описания **CSS**

- .menu {
- font-weight: bold;
- font-size: 9pt;
- font-family: arial, helvetica, sans-serif;
- }
- a:hover {
- color: #b63a3a;
- text-decoration: none;
- }

примеры описания **CSS**

- Эти примеры помогут Вам на первых порах, а для написания более сложных стилевых таблиц лучше всего воспользоваться специальными программами - например, входящим в состав программы HomeSite редактором TopStyle, который, кстати, содержит и прекрасный справочник свойств.