

Каскадные таблицы стилей лаба

Цель работы: формирование умений создания HTML-документов с применением каскадных таблиц стилей.

Для чего нужны таблицы стилей?

Вы создавали предыдущие страницы, так как их создавали раньше до появления каскадных таблиц стилей или CSS (Cascading Style Sheets).

Основные проблемы, с которыми сталкивались разработчики сайтов до появления CSS:

- Проблема позиционирования (разметка страниц). Хотя для этого стали использовать таблицы (они небыли для этого предназначены изначально), они не решили всех проблем. Не возможно осуществить следующее:
 - Нельзя задать фиксированные размеры какого-нибудь блока. Например, текстовый блок будет разезжаться при переполнении (даже если он в таблице).
 - Нельзя задать фиксированные координаты положения блока на странице.
 - Нельзя наложить один блок на другой. Например, попробуйте надвинуть картинку поверх таблицы или другой картинки.
- Вторая проблема заключалась в том, что приходилось каждый раз задавать на страницах размер и цвет шрифта, свойства ячеек таблицы и т.д. Это сильно увеличивало размеры страницы, а значит, она медленнее загружается. В случае если вам захотелось изменить цвет или размер шрифта, вам пришлось бы редактировать все страницы.
- Третья проблема заключалась в том, что все браузеры имеют индивидуальные настройки. Например, в браузере можно увеличить шрифт, что приведет к искажению всей страницы.

С помощью CSS эти проблемы можно решить.

Способы применения CSS

Существует три способа применения таблиц стилей:

- **Внутренние таблицы стилей (Inline Style Sheets)** - при помощи специального атрибута помещаются прямо в HTML теги.

Пример HTML:

```
<font color="blue" size="3" face="Arial"> Пример </font>
```

Пример CSS:

```
<font style="color:blue; font-size:12pt; font-family:Arial"> Пример </font>
```

Как можно заметить, код Inline Style Sheet получился больше чем HTML. Поэтому ISS следует использовать, только если необходимо задать определенному элементу свой индивидуальный стиль. 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. Но представьте, насколько вырастет размер файла, и насколько неудобно будет его исправлять, если мы будем указывать стиль у каждого тега.

- **Глобальные таблицы стилей (Global Style Sheets)** - определяют стиль элементов во всем документе.

Для этого используется тег `<STYLE type="text/css">`. Он размещается в заголовке документа `<head>`

Пример:

```
<html>
<head>
<STYLE type="text/css">
h1 { color:red; font-style:italic; font-size:32px} - переопределение стандартного тега.
.blue { color:blue; } - определение нового класса
</STYLE>
</head>
<body>
```

Теперь эти стили можно применять в любом месте html-кода. Для этого используются следующие конструкции:

```
<h1> Этот заголовок написан крупным красным курсивом </h1>
Вот <font class="blue"> это </font> слово - синее.
</body>
</html>
```

В данном примере все элементы H1 будут написаны крупным красным курсивом, все элементы с указанным классом BLUE будут синими.

- **Связанные таблицы стилей** (Linked Style Sheets) - могут быть использованы для нескольких документов сразу и хранятся во внешнем файле.

Пример внешнего файла:

Файл main.css

```
body {background:black; font-size:9pt; color:red; font-family:Arial Black}
.base{color:blue; font-style:italic}
h1 {color:white}
#bold { font-weight:bold}
```

В самих же HTML документах делается ссылка на этот файл при помощи тега <LINK> Выглядит это так:

```
<LINK rel="STYLESHEET" TYPE="text/css" HREF="путь до файла">
```

Пример:

Файл Index.html

```
<html>
<head>
<LINK rel="stylesheet" type="text/css" href="main.css">
</head>
<body>
Содержание Документа
</body>
</html>
```

Это самый удобный способ, и для основной таблицы стилей рекомендуется пользоваться именно им. Почему каскадные?

Каскадность заключается в том, что стили могут переопределяться. Приведенный ниже список способов внедрения стилей соответствует порядку переопределения. Нижерасположенный способ может переопределять вышерасположенный.

Способы внедрения стилей на странице:

- использование отдельного стилевого файла и вставка его при помощи тега <link>
- описание стиля в заголовке документа
- применения стиля как параметра в теге.

Например, мы определили во внешнем стилевом файле, что текст в теге <p> должен быть написан при помощи шрифта высотой 10 пунктов. Но если в заголовке странички мы дополнительно укажем, что тот же текст в теге <p> должен быть написан шрифтом в 12 пунктов, то текст будет выведен шрифтом 12 пунктов - т.е. стиль в заголовке странички переопределил стиль во внешнем файле.

Для чего это нужно? Предположим, что для всех ссылок в заголовке на нашей страничке определен следующий стиль:

```
<style type="text/css">
<!--
```

```
a.link { text-decoration: none; color:red; }
```

```
-->
</style>
</head>
```

И, следовательно, любой текст, который является гиперссылкой, автоматически становится красным и перестает быть подчеркнутым. И вот мы решили в конце странички указать ссылку на сайт автора странички, но сделать это не сильно заметно, чтобы не акцентировать на этом внимание. Сделать это нам нужно всего лишь в одном месте странички и определять для этого дополнительный класс нецелесообразно. Достаточно локально переопределить цвет ссылки:

```
<a href="http://www.pupkin.ru"><span style="color: #000000;">Вася Пупкин</span></a>
или так
<a href="http://www.pupkin.ru" style="color: #000000;">Вася Пупкин</a>
```

Сделали мы это при помощи атрибута style, а он действует лишь в пределах того тега, в котором был определен.

Тег `` определяет некую область, к которой мы можем применить стиль.

Наследование

Некоторые значения наследуются дочерними элементами (тегами).

Предположим, что имеется элемент H1, в котором расположен элемент выделения EM:

```
<H1>Данный заголовок <EM>очень</EM> важен!</H1>
```

Если для элемента EM не задан цвет, то слово "очень" унаследует цвет родительского элемента. Таким образом, если для H1 определен синий цвет, то и элемент EM будет представлен, синим цветом.

Определение свойств стиля, используемого в документе по умолчанию, можно осуществить в корневом элементе дерева данного документа. Например, в языке HTML эту возможность можно реализовать с помощью элементов HTML или BODY.

Пример(ы):

Например, свойство 'color' является наследуемым, поэтому все потомки элемента BODY унаследуют цвет 'black':

```
BODY { color: black; }
```

Значения, заданные в процентах, не наследуются, а вычисляемые значения наследуются.

Пример(ы):

Пусть задана следующая таблица стиля:

```
BODY { font-size: 10pt }
```

```
H1 { font-size: 120% }
```

и фрагмент документа:

```
<BODY>
<H1>Некоторый <EM>крупный</EM> заголовок</H1>
</BODY>
```

Свойство 'font-size' элемента H1 будет иметь вычисленное значение '12pt' (120% от 10pt, являющегося значением свойства родительского элемента). Так как вычисляемое значение свойства 'font-size' является наследуемым, то элемент EM также будет иметь вычисленное значение '12pt'.

Синтаксис CSS

Все объявления CSS (официально они называются "селекторы") записываются в фигурных скобках:

ТЕГ: псевдокласс.класс { характеристика1: значение1; характеристика2: значение2; }

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; }
```

Универсальный класс - может быть применен к любому тегу, имя класса начинается с точки.

Пример:

```
<p class=small>Накладываем стиль на этот текст</p>
```

```
<td class=small>Накладываем стиль на этот текст</td>
```

Теговые классы - может быть применен к конкретному тегу, имя класса начинается с указания тега, после точки записывается имя класса.

Пример:

```
p.small { font-size: 9pt; }
```

Класс, определенный таким образом, сработает только в том теге, для которого он предназначен, а для всех остальных будет проигнорирован.

```
<p class=small>Этот текст будет выведен стилем small</p>
```

```
<td class=small>А этот останется неизменным</td>
```

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, LI { font-size: 9pt; color: green; }
```

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, и его можно наблюдать в том числе и на этом сайте. Вот фрагмент таблицы стилей, который позволяет достигать вышеописанного эффекта:

```
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.

А вот и другой пример псевдокласса - определение буквицы в начале 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.

Пример:

```
/* 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.

Некоторые параметры CSS

Некоторые 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

Некоторые 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

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), средней шириной буквы "m" (em), средней шириной буквы "x" (ex)
- 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),...

Пример использования CSS

Назначим для всех состояний ссылок цвета и сделаем так, чтобы ссылки во всех состояниях не имели подчеркивания, а в момент нахождения курсора над ссылкой появлялось подчеркивание.

```
A:link {COLOR: ; text-decoration: none}
```

```
A:visited {COLOR: black; text-decoration: none}
```

```
A:active {COLOR: red; text-decoration: none}
```

```
A:hover {COLOR: #6666FF; text-decoration: underline}
```

- ссылка (link) - синий цвет (#0000CC);

- посещенная ссылка (visited) - черный цвет (black);
- активная ссылка (active) - красный цвет (red);
- ссылка, в момент нахождения курсора мыши над ней (hover) - светло-синий (#6666FF);

Используя CSS, мы можем отменить или переназначить эффект для ссылок. Для этого используем свойство text-decoration, которое может иметь следующие значения:

none - без оформления
 underline - подчеркивание
 overline - черта сверху
 line-through - перечеркивание
 blink - мигание

Практические задания

Задание 1

1. Создайте каталог /css/
2. Создайте в нем файл main.css
3. Подключите его ко всем страницам сайта с помощью тега <LINK>.
4. Задайте по умолчанию следующие параметры для всех страниц (перопределив, тег <body>):
 - цвет фона
 - размер шрифта
 - цвет шрифта
 - семейство шрифта (например, Arial)
5. Проверьте работоспособность на сервере.
6. В комментариях (в файле main.css) поясните эти параметры.

Задание 2

1. Задайте по умолчанию следующие параметры для всех абзацев (перопределив, тег <p> и псевдоклассы тега <p>):
 - выравнивание абзаца
 - отступ красной строки
 - размер и цвет первой буквы
2. Проверьте работоспособность на сервере.
3. В комментариях (в файле main.css) поясните эти параметры.

Задание 3

1. Задайте по умолчанию следующие свойства ссылок для всех страниц:
 - цвет и оформление ссылки
 - цвет и оформление посещенной ссылки
 - цвет и оформление активной ссылки
 - цвет и оформление ссылки, в момент нахождения курсора мыши над ней
2. Проверьте работоспособность на сервере.
3. В комментариях (в файле main.css) поясните эти параметры.

К соаче лабораторной предоставляются: работающие страницы на сервере.