# Лабоработрная работа №3 Каскадные таблицы стилей CSS

***Цель работы***: изучить и научиться использовать на практике каскадные таблицы стилей CSS.

***Задание***

Отредактировать HTML-страницу, созданную в прошлой лабораторной работе. Убрать из HTML-документа все теги и атрибуты, связанные с форматированием страницы. Разработать CSS-файл для документа, куда поместить всё форматирование. Использовать классы, идентификаторы и следующие правила CSS:

Параметры текста: *Color Direction, Letter-spacing, Text-align, Text-decoration, Text-indent, Text-transform, White-space, Word-spacing* .

Параметры фона: Background, Background-attachment, Background-color, Background-image, Background-position, Background-repeat.

Параметры границ: Border, Border-collapse, Border-color, Border-style, Border-width, Border-spacing.

Параметры шрифта: Font, Font-size, Font-family, Font-style, Font-variant, Font-weight .

Параметры списков: List-style, List-style-image, List-style-position, List-style-type.

***Методические указания к выполнению работы***

**CSS** (*Cascading Style Sheets* — ***каскадные таблицы стилей***) — технология описания внешнего вида документа, написанного языком разметки.

Преимущественно используется как средство оформления веб-страниц в формате HTML и XHTML, но может применяться с любыми видами документов в формате XML, включая SVG и XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления документа. Основной целью разработки CSS являлось разделение содержимого (написанного на HTML) и представления документа (написанного на CSS). Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печать, и даже чтение голосом.

***Понятие стиля, таблицы стилей и CSS***

Стиль представляет собой набор параметров, определяющих внешний вид документа HTML при его отображении в окне браузера: шрифты и цвета заголовков разных уровней, шрифт и разрядка основного текста, задаваемого в тэге абзаца <p>, и т. д. Стиль задается по определенным правилам

Таблица стилей - набор правил отображения, применяемых в документе, к которому присоединена соответствующая таблица стилей. Таблица стилей — это шаблон, который управляет форматированием тэгов HTML в Web-документе.

Термин "каскадные" используется потому, что возможно использовать несколько таблиц стилей для управления форматированием одного документа HTML, а браузер по определенным правилам выстраивает приоритетность применения этих таблиц.

Применение технологии CSS обеспечивает:

* Гибкое размещение информации о стиле. Помещение таблиц стилей в отдельные файлы упрощает их повторное использование. Иногда полезно включать инструкции по представлению в документ, к которому они применяются, в начало документа или в атрибуты элементов в теле документа.
* Каскады. Эта возможность обеспечивается некоторыми языками таблиц стилей, такими как CSS, для объединения информации о стиле из нескольких источников. Это может быть, например, корпоративные положения о стиле, стили, общие для группы документов, а также стили, специфичные для одного документа. С использованием раздельного хранения эти таблицы стилей могут использоваться повторно, что упрощает работу авторов и повышает эффективность сетевого кэширования. Каскад определяет упорядоченную последовательность таблиц стилей, в которой правила более поздних таблиц имеют приоритет над более ранними.
* Зависимость от устройств. HTML позволяет авторам разрабатывать документы, независимые от устройств. Это позволяет пользователям обращаться к Web-страницам с использованием различных устройств, например, графических дисплеев для компьютеров под управлением различных ОС, телевизионных устройств, телефонов и портативных устройств на базе PDA, речевых браузеров и тактильных устройств на базе азбуки Бройля. Таблица стилей, напротив, применяются к конкретным устройствам или группам устройств. Языки таблиц стилей могут включать функции описания зависимости от устройств в одной таблице.
* Альтернативные стили. Авторы могут предлагать читателям несколько способов просмотра документа. Например, таблица стилей для представления компактных документов с мелким шрифтом, или таблица, задающая крупные шрифты для удобства чтения. Автор может указать предпочитаемую таблицу стилей, а также альтернативные таблицы для определенных пользователей или устройств. Агенты пользователей должны предоставлять пользователям возможность выбора одной из альтернативных таблиц или отключать все таблицы стилей.
* Вопросы производительности. Загрузка внешней таблицы стилей может привести к задержке общего представления материала для пользователя. Подобные ситуации возникают и в том случае, если в заголовок документа включен длинный набор правил относительно стиля. При использовании общей таблицы стилей для группы документов снижение производительности произойдет только для первого документа, а для остальных документов таблицы стилей уже будет находиться в локальном кэше.

CSS при отображении страницы может быть взята из различных источников:

* Авторские стили (информация стилей, предоставляемая автором страницы) в виде:
  + Внешних таблиц стилей, то есть отдельного файла .css, на который делается ссылка в документе.
  + Встроенных стилей — блоков CSS внутри самого HTML-документа.
  + Inline-стилей, когда в HTML-документе информация стиля для одного элемента указывается в его атрибуте style.
* Пользовательские стили
  + Локальный CSS-файл, указанный пользователем в настройках браузера, переопределяющий авторские стили, и применяемый ко всем документам.
* Стиль [браузера](http://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80)
  + Стандартный стиль, используемый браузером по умолчанию для представления элементов.

***Правила CSS***

Таблица стилей состоит из набора правил.

Правило каскадных таблиц стилей состоит из двух частей: селектора и определения. Селектором может быть любой тэг HTML, для которого определение задает, каким образом необходимо его форматировать. Определение располагается в правой части правила. Оно помещается в фигурные скобки. Само определение, в свою очередь, также состоит из двух частей: свойства и его значения, разделенных знаком двоеточия (:). ,Пример:

h1 {color: blue; font-size: 16pt}

Назначение свойства очевидно из его названия. В приведенном правиле селектором является элемент h1, а определение, записанное в фигурных скобках, задает значения двух свойств заголовка первого уровня: цвет шрифта (свойство color) определен как синий (значение blue) и размер шрифта (свойство font-size) определен в 16 пунктов (значение 16pt). В одном правиле можно задать несколько определений, разделенных символом точка с запятой (;).

Правило CSS, может задаваться с помощью атрибута style элемента, к которому оно применяется.

Для уменьшения размеров таблиц стилей можно группировать разные селекторы в виде списка элементов страницы HTML, разделенных запятыми, если для них задается одно правило. Например, следующие правила:

h1 {font-family: Arial}

h2 {font-family: Arial}

h3 {font-family: Arial}

можно сгруппировать и задать в виде одного правила со списком селекторов:

h1,h2,h3 {font-family: Arial}

Аналогично группируются определения, только в списке они разделяются точками с запятой (;). Следующие правила форматирования заголовка первого уровня:

h1 {font-weight: bold}

h1 {font-size: 14pt}

h1 {font-family: Arial}

можно задать в виде одного правила, сгруппировав определения:

h1 {font-weight: bold; font-size: 14pt; font-family: Arial}

Некоторые свойства имеют собственный синтаксис группирования, связанный с заданием значений нескольких свойств в одном. Например, предыдущий пример при использовании свойства font запишется так:

h1 {font: bold 14pt Arial}

При задании таблицы стилей можно свободно комбинировать все три правила группирования для уменьшения ее размеров.

*Наследование*

В HTML некоторые элементы могут содержать другие. Рассмотрим элемент, расположенный внутри другого элемента страницы, если для последнего задано правило форматирования, а для вложенного элемента нет.

Например, пусть цвет шрифта абзаца определен как синий (p {color:blue}). Рассмотрим выделенный элемент текста, задаваемый тэгом <em>, если для него не определено правило форматирования.

В подобных случаях вложенный элемент наследует правила форматирования элемента-родителя.

Наследование полезно при задании значений свойств, применяемых к документу по умолчанию. Для этого достаточно задать все свойства для элемента, порождающего все остальные элементы страницы HTML. Таким элементом является тело документа, определяемое тэгом <body>:

body {

color: black;

font-family: "Times New Roman";

background: url(texture.gif) white;

}

*Селекторы*

Правила каскадных таблиц стилей, в которых в качестве селектора используются тэги HTML, влияют на отображение всех элементов заданного типа в документе. Следующее правило отображает без подчеркивания все ссылки (тэг <a>) в документе:

<style type="text/css">

a { text-decoration: none; }

</style>

Если необходимо некоторые ссылки отобразить по-другому, то можно задать для них правило форматирования непосредственно в тэге, а можно применить параметр class, добавленный в HTML 4.0 в качестве стандарта для всех тэгов. Значением параметра class является ссылка на класс, задаваемый в таблице стилей.

*Селектор class*

В одной таблице стилей можно создать два и более различных класса одинаковых элементов с помощью селектора класса. К этим классам элементов можно затем будет применить различные таблицы стилей.

Пример. Необходимо создать страницу, на которой будет два вида абзацев <p>, причем оба вида будут постоянно чередоваться и часто повторяться. Типичный пример такой страницы – интервью, в котором чередуются вопросы журналиста и ответы интервьюируемого. При создании такой страницы необходимо визуально отделить вопросы и ответы друг от друга.

<head>

<title>Интервью</title>

<style>

p.ask {

font-style: italic;

font-weight: bold;

font-family: Arial, sans-serif;

font-size: 10pt;

color: gray;

margin-left: 15px

}

p.answer {

font-family: 'Times New Roman', serif;

font-size: 12 pt; color: black;

}

</style>

</head>

<body>

<p class="ask">Вопрос</p>

<p class="answer">Ответ</p>

</body>

*Селектор id*

Параметр id, задает уникальное имя элемента, которое используется для ссылок на него в сценариях и таблицах стилей. Параметр id можно применять к любому элементу документа.

Правила таблиц стилей регламентируют использование уникального идентификационного имени элемента в качестве селектора, предваряя его символом #:

<head>

<title>Демо</title>

<style type="text/css">

<!- #par24 { letter-spacing: lem; }

h1#form3 { color: red; background-color: blue}

-->

</style>

</head>

<body>

<p id="par24">Разреженные слова в абзаце</p>

<h1 id="form2">Черный шрифт</h1>

</body>

В этом примере абзац идентифицирован именем par24 в параметре id, поэтому к нему применимо правило с селектором #par24. Второе правило в таблице стилей должно применяться к заголовку первого уровня с идентификатором form3. Такого элемента в нашем фрагменте нет, и поэтому заголовок form2 отображается с применением правила по умолчанию.

Пример таблицы стилей:

p {

font-family: "Garamond", serif;

}

h2 {

font-size: 110 %;

color: red;

background: white;

}

.note {

color: red;

background: yellow;

font-weight: bold;

}

p#paragraph1 {

margin: 0;

}

a:hover {

text-decoration: none;

}

#news p {

color: blue;

}

Здесь приведено шесть правил с селекторами p, h2, .note, p#paragraph1, a:hover и #news p.

В первых двух правилах HTML-элементам p (абзацу) и h2 (заголовку второго уровня) назначаются стили. Абзацы будут отображаться шрифтом Garamond, или, если такой шрифт недоступен, каким-либо другим шрифтом с засечками («serif»). Заголовок второго уровня будет отображаться красным на белом фоне с увеличенным кеглем.

*Связывание HTML с CSS*

Чтобы таблица стилей могла воздействовать на внешнее представление документа, браузер должен знать о ее существовании. Для этого ее необходимо связать с HTML-документом. Существует четыре способа связывания документа и таблицы стилей:

1. Связывание — позволяет использовать одну таблицу стилей для форматирования многих страниц HTML.
2. Внедрение — позволяет задавать все правила таблицы стилей непосредственно в самом документе.
3. Импортирование — позволяет встраивать в документ таблицу стилей, расположенную на сервере.
4. Встраивание в тэги документа — позволяет изменять форматирование конкретных элементов страницы.

*Связывание таблицы стилей*

Связывание позволяет хранить таблицу стилей в отдельном файле и присоединять ее к документам с помощью тэга <link>, задаваемого в разделе <head>:

<link rel="stylesheet" type="text/css" href="mystyles.css"/>

В этой строке указывается, что связываемый файл является таблицей стилей (rel="stylesheet"), формат этого файла – .css1 (type="text/css") и находится он в той же директории, что и файл html, либо имеет другой URL-адрес (href="mystyles.css").

Эта строка может быть указана в любом html-файле. Таким образом, единое оформление будет использовано для нескольких страниц.

При этом вся таблица стилей хранится в одном файле (расширение файла должно быть стандартным – .css).

Пример CSS-файла:

body {

font-family: 'Times New Roman';

font-size: 12pt;

color: green;

}

h1 {

font-family: Arial;

font-size: 16pt;

color: blue;

font-weight: bold;

}

h2 {

font-family: Helvetica;

font-size: 14pt;

color: yellow;

font-weight: bold;

font-style: italic;

}

Тэги <style> и </style> внутри файла таблицы стилей не используются – расширение .css явно указывает браузеру на то, что файл является таблицей стилей. Один такой файл может быть связан сразу с несколькими страницами (или импортирован сразу в несколько страниц).

*Внедрение таблицы стилей*

Для применения одинакового форматирования к нескольким одинаковым элементам станицы необходимо создать в заголовке страницы (в любом месте между тегами <head> и </head>) внедренную таблицу стилей, в которой задаются требуемые правила оформления. Для этого создается тег-контейнер таблицы стилей, начинающийся открывающим тегом <style> и заканчивающийся закрывающим тегом </style>. Внутри этого тега-контейнера можно задать любое количество правил CSS, состоящих из селектора (названия тега HTML, к которому будет применяться правило) и его определения (непосредственно набора средств форматирования), заключенного в фигурные скобки.

Пример. Необходимо, чтобы все абзацы на странице выглядели, как в предыдущем примере, все заголовки первого уровня отображались шрифтом Arial синего цвета полужирного начертания размером 16 пунктов, а все заголовки второго уровня – шрифтом Helvetica размером 14 пунктов полужирного курсивного начертания желтого цвета.

<head>

...

<style>

<!-body {

font-family: 'Times New Roman';

font-size: 12pt;

color: green;

}

h1 {

font-family: Arial;

font-size: 16pt;

color: blue;

font-weight: bold;

}

h2 {

font-family: Helvetica;

font-size: 14pt;

color: yellow;

font-weight: bold;

font-style: italic;

}

-->

</style>

...

</head>

Этот способ связывания CSS и HTML называется внедрением. Его рекомендуется применять в тех случаях, когда необходимо задать какой-либо набор правил форматирования только для одной страницы сайта, а все остальные страницы должны выглядеть по-другому.

*Импорт таблицы стилей*

Для импортирования файла таблицы стилей (в том числе с другого сервера) необходимо указать в заголовке HTML-файла между тегами <head> и </head> внутри тега-контейнера <style> обращение к файлу таблицы стилей:

<head>

...

<style>

...

@import: url(my.css);

...

</style>

...

</head>

Помимо адреса импортируемой таблицы стилей, в теге-контейнере <style> можно указать любые правила CSS, которые будут дополнять или корректировать правила, заданные в импортируемой таблице – внедренные правила.

*Встраивание CSS в теги документа*

CSS позволяют назначить собственный стиль визуального представления любому тегу HTML, в том числе тегу <body>. Если стиль задан для тега <body>, он наследуется всеми элементами (абзацами, заголовками и т. д.), помещенными внутри этого тега-контейнера, в случае отсутствия собственных стилей для этих элементов. Таким образом, нет необходимости прописывать теги <font> и атрибуты color, size и т. п. для каждого абзаца на странице – достаточно задать стиль для тега <body>, и все абзацы на странице будут отображены в соответствии с этим стилем.

Пример. Необходимо, чтобы все абзацы отображались шрифтом Times New Roman размером 12 пунктов зеленого цвета. Для этого следует указать атрибут style тега <body>, присвоив ему соответствующее значение:

<body style="font-family: 'Times New Roman'; font-size: 12pt; color: green">

В примере используется встраивание стиля непосредственно в тег документа – так называемый inline-стиль. Этот способ связывания CSS с HTML-файлом рекомендуется в единичных случаях – если данный стиль планируется применить только к одному элементу только на одной странице сайта. Если же стиль должен быть применен к нескольким элементам на одной странице или к нескольким страницам сразу, рекомендуются другие способы связывания CSS и HTML

*Приоритеты использования таблиц стилей*

Браузер расставляет приоритеты таблиц стилей следующим образом:

1. встроенные (inline-) стили (встроенные с помощью атрибута style непосредственно в теги документа) – наивысший приоритет. Будут применены браузером в любом случае, даже если возникает конфликт с внедренными или внешними стилями;
2. внедренные стили (перечисленные в теге-контейнере <style> в заголовке документа) – чуть меньший приоритет, будут применяться во всех случаях, кроме случаев возникновения конфликта с inline-стилями (при возникновении такого конфликта будут применены inline-стили);
3. импортированные стили (стили внешнего файла .css, связанные с документом с помощью свойства @import в теге-контейнере <style>) – будут применяться в тех случаях, когда отсутствуют аналогичные правила CSS среди встроенных и внедренных стилей;
4. связанные стили (стили, присоединенные к html-файлу посредством тега <link>) – наименьший приоритет, будут применены только после того, как браузер убедится в отсутствии аналогичных правил во всех остальных типах стилей.

***Контрольные вопросы***

1. Для чего используется технология CSS?
2. Начиная с какой версии HTML была реализована технология CSS?
3. Какую смысловую нагрузку несёт термин "каскадные" в технологии CSS?
4. Какие преимущества обеспечивает технология CSS?
5. Из каких источников может быть взята таблица CSS при отображении WEB-страницы браузером?
6. Как задаётся правило CSS?
7. Как задаётся наследование в CSS?
8. Как можно задать селекторы?
9. Как можно связать HTML-документ с таблицей стилей?
10. Приоритеты использования таблиц стилей.

***Список источников***

# Введение в CSS <http://uroki-css.ru/css/css_examples.php>

# Справочник CSS <http://htmlbook.ru/css>

# Примеры CSS <http://uroki-css.ru/css/css_examples.php>

# [Все спецификации CSS](Все%20спецификации%20CSS%20) <https://www.w3.org/Style/CSS/specs.ru.html>