8. Каскадные таблицы стилей CSS

Цель: познакомиться с каскадными таблицами стилей CSS, их достоинствами и недостатками; определением правил стилей и селекторами.

8.1 Теоретические сведения

В то время как HTML структурирует документ и сообщает веб-браузеру, какую функцию имеет определенный элемент, CSS выдает браузеру инструкции о том, как отобразить определенный элемент - оформление, размещение пробелов и позиционирование.

CSS - одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название "стандарты Web ".

начало 1990-х годов	Различные браузеры имели свои стили для отображения веб страниц. HTML развивался очень быстро и был способен удовлетворить все существовавшие на тот момент потребности по оформлению информации, поэтому CSS не получил тогда широкого признания
1994	Появился термин "каскадные таблицы стилей"
1996	Консорциумом W3C была издана рекомендация *CSS1*
1998	Консорциумом W3C принята рекомендация *CSS2*
Сентябрь 2009	Консорциумом W3C утверждена рабочая версия *CSS2.1*

8.1.1 Уровень 1 (CSS1)

Эта рекомендация W3C была принята в 1996 году и откорректирована в 1999 году. Основные возможности, предоставляемых этой рекомендацией:

- Параметры шрифтов. Возможности по заданию гарнитуры и размера шрифта, а также его стиля - обычного, курсивного или полужирного.
- Цвета. Спецификация позволяет определять цвета текста, фона, рамок и других элементов страницы.
- Атрибуты текста. Возможность задавать межсимвольный интервал, расстояние между словами и высоту строки (то есть межстрочные отступы)
- Выравнивание для текста, изображений, таблиц и других элементов.
- Свойства блоков, такие как высота, ширина, внутренние (padding) и внешние (margin) отступы и рамки. Также в спецификацию входили ограниченные средства по позиционированию элементов, такие как float и clear.

8.1.2 Уровень 2 (CSS2)

Эта рекомендация W3C была принята в 1998 году. Она построена на CSS1 с сохранением обратной совместимости. Среди новых возможностей можно назвать следующие:

- Блочная верстка. Появились относительное, абсолютное и фиксированное позиционирование. Позволяет управлять размещением элементов по странице без табличной верстки.
- Типы носителей. Позволяет устанавливать разные стили для разных носителей (например монитор, принтер, КПК).
- Звуковые таблицы стилей. Определяет голос, громкость и т. д. для звуковых носителей (например для слепых посетителей сайта).
- Страничные носители. Позволяет, например, установить разные стили для элементов на четных и нечетных страницах при печати.
- Расширенный механизм селекторов.
- Указатели.
- Генерируемое содержание. Позволяет установить текст или картинку, который будет отображаться до или после нужного элемента.

8.1.3 Уровень 2.1 (CSS 2.1)

Рабочая версия W3C от 8 сентября 2009 года. Построена на CSS2, в ней исправлен ряд ошибок.

8.1.4 CSS-верстка

Использование CSS дает следующие преимущества:

- Несколько дизайнов страницы для разных устройств просмотра. Например, на экране дизайн будет рассчитан на большую ширину, во время печати меню не будет выводиться, а на КПК и сотовом телефоне меню будет следовать за содержимым.
- Уменьшение времени загрузки страниц сайта за счет переноса правил представления данных в отдельный CSS -файл. В этом случае браузер загружает только структуру документа и данные, хранимые на странице, а представление этих данных загружается браузером только один раз и кешируется.
- Простота последующего изменения дизайна. Не нужно править каждую страницу, а лишь изменить CSS -файл.
- Дополнительные возможности оформления. Например, с помощью CSS -верстки можно сделать блок текста, который остальной текст будет обтекать (например для меню) или сделать так, чтобы меню было всегда видно при прокрутке страницы.

Известны также и недостатки:

- Различное отображение верстки в различных браузерах (особенно устаревших), которые по разному интерпретируют одни и те же данные CSS.
- Часто встречающаяся необходимость на практике исправлять не только один CSS -файл, но и теги HTML и код PHP, которые сложным и ненаглядным способом связаны с селекторами CSS, что иногда сводит на нет простоту применения единых файлов стилей и значительно удлиняет время редактирования и тестирования.

Отображение элементов реализуется с помощью системы правил, которые определяют, какие элементы HTML должны быть дополнительно оформлены, и в каждом правиле перечислятся свойства (например, цвет, размер, шрифт, и т.д.) этих элементов HTML, которыми они будут манипулировать, какие значения будут для них заданы.

Таким образом, CSS не является ни языком программирования, ни языком разметки.

8.1.5 Определение правил стилей

Базовой конструкцией в CSS является правило следующего вида:

```
селектор {
  свойство1: значение;
  свойство2: значение;
  свойство3: значение;
}
```

Селектор определяет элементы HTML, к которым будет применяться правило, определяемые реальным названием элемента, например, body, или другими средствами, такими как значения атрибута class.

Фигурные скобки {} содержат пары свойство/значение, которые разделяются между собой точкой с запятой; свойства отделяются от своих соответствующих значений двоеточием.

Свойства определяют, что вы хотите сделать с выделенными элементами. Они могут задавать, например, цвет элемента, цвет фона, позицию, поля, заполнение, тип шрифта, и многое другое.

Значения являются теми конкретными характеристиками, которые вы хотите задать каждому свойству выделенных элементов. Эти значения зависят от свойства.

Свойства, которые влияют на положение, поля, ширину, высоту и т.д. могут измеряться в пикселях, em, процентах, сантиметрах или других аналогичных единицах измерения.

Рассмотрим конкретный пример:

```
p {
  margin: 10px;
  font-family: Times New Roman;
  color: green;
}
```

```
<html>
  <head>
      <style type="text/css">
          p { color:Maroon;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>
</html>
```

Это правило выбирает HTML элемент p, и для каждого элемента p в документах HTML, которые используют этот код CSS, будет применяться это правило, если только не будут существовать применяемые к ним более конкретные правила, которые будут переопределять это правило. Данное правило влияет на свойства, которые определяют поля вокруг параграфа (margin), шрифт текста в параграфе (font-family), и цвет этого текста (color). Поля задаются размером в 10 пикселей, шрифт задается как Times New Roman, а цвет текста задается как green.

Все множество подобных правил совместно формируют таблицу стилей. Кроме таких правил в CSS существуют и другие конструкции, например комментарии CSS, селекторы объединения в группу.

8.1.6 Комментарии CSS

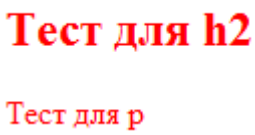
Комментарии добавляют, помещая их между символами /* и */. Комментарии могут охватывать несколько строк, и в этом случае браузер будет игнорировать эти строки.

8.1.7 Селекторы объединения в группу

Можно также объединить в группу различные селекторы. Предположим, что вы хотите применить одинаковое оформление к h2 и p, тогда можно было бы написать следующий CSS:

```
<STYLE>
   h2 {color: red}
   p {color: red}
</STYLE>

<html>
   <head>
   <style type="text/css">
      h2 {color: red}
      p {color: red}
   </style>
   </head>
   <body>
   <h2>Тест для h2</h2>
            <p>
                     Тест для p
            </p>
   </body>
</html>
```



Однако, можно сократить код CSS, группируя селекторы вместе с помощью запятой.

Существует несколько различных селекторов, каждый из которых соответствуют различным частям разметки. Тремя наиболее общими селекторами, которые встречаются чаще всего, являются следующие:

Синтаксис	Селектор	Описание
p {}	селектор элемента	соответствует всем элементам на странице с указанным названием (элементам p, в приведенном выше случае)
example{}	селектор класса	соответствует всем элементам, которые имеют атрибут class с указанным значением, так что пример выше будет соответствовать <p class="example">, <li class="example"> или , или любому другому элементу со значением class = "example".
#example{}	селектор id	соответствует всем элементам, которые имеют атрибут id с указанным значением, так что пример выше будет соответствовать <p id="example">, <li id="example"> или , или любому другому элементу со значением id = "example". Селекторы id не проверяют название никакого элемента, и можно иметь только один селектор для каждого id в документе HTML - они являются уникальными для каждой страницы.

8.1.8 Дополнительные селекторы CSS

С помощью селекторов CSS, селекторов элемента, класса и id -селекторов можно реализовать многое, но это, конечно, не все возможные селекторы - существуют другие селекторы, которые позволяют выбирать элементы для стилевого оформления на основе более специфических критериев.

8.1.9 Универсальные селекторы

Универсальные селекторы выбирают каждый элемент на странице для применения к ним стилей оформления.

8.1.9.1 Селекторы атрибутов элементов

Селекторы атрибутов позволяют выбирать элементы на основе содержащихся в них атрибутов.

8.1.9.2 Селекторы потомков элементов

Можно использовать селектор потомка для выбора только определенных элементов, которые являются потомками других определенных элементов.

8.1.9.3 Селекторы нижележащих элементов

Селекторы нижележащих элементов очень похожи на селекторы потомков, за исключением того, что селекторы потомков выбирают только непосредственно нижележащих, а селекторы нижележащих выбирают подходящие элементы в любом месте иерархии элементов, а не только непосредственно нижележащих.

8.1.9.4 Селекторы смежных одноуровневых элементов

Эти селекторы позволяют выбирать определенный элемент, который следует непосредственно после другого определенного элемента на том же уровне в иерархии элементов.

8.1.9.5 Псевдо-классы

Псевдо-классы используются для обеспечения стилевого оформления не для элементов, а для различных состояний элементов. Наиболее обычным применением, которое можно встретить, является оформление состояний ссылок.

Список ниже представляет различные псевдо-классы и описание состояния ссылки, которое они выбирают:

:link	обычное состояние ссылок по умолчанию, когда вы впервые их находите
:visited	ссылки, которые вы уже посетили в используемом в данный момент браузере
:focus	ссылка (или поле формы, или что-то еще), в которой в данный момент находится курсор клавиатуры
:hover	ссылка, на которой в данный момент находится указатель мыши
:active	ссылка, на которой в данный момент происходит щелчок

Следующие правила CSS определяют что:

- по умолчанию ссылки будут синими.
- когда курсор мыши оказывается над ссылкой, используемое по умолчанию подчеркивание ссылки исчезает.
- когда ссылка будет посещена, она станет серой.
- когда ссылка активна, она становится жирной, как дополнительный признак, что что-то сейчас произойдет.

8.1.9.6 Псевдо-элементы

Псевдо-элементы имеют два назначения. Прежде всего, можно использовать их для выбора первой буквы или первой строки текста в заданном элементе.

Вторым применением псевдо-элементов является генерация контента с помощью CSS, что является более сложной задачей. Можно использовать псевдо-элементы :before или :after для определения того, что содержимое должно быть вставлено перед или после элемента, который вы выбираете. Затем определяют то, что должно быть вставлено.

8.1.10 Сокращенная запись CSS

Можно объединить несколько связанных свойств CSS в одно свойство, чтобы сэкономить время и свои усилия.

8.1.10.1 Задание менее четырех значений для сокращенного свойства

Сокращенное значение может иметь менее четырех значений, согласно приведенному ниже списку. Результаты упорядочены по числу предоставленных значений:

Одно значение применяется ко всем четырем сторонам	margin: 2px;
Первое значение применяется к верху и низу, *второ* е к левому и правому краю	margin: 2px 5px;.
Первое и *третье* значения применяются к верху и низу соответственно, *второе* значение применяется к левому и правому краю	margin: 2px 5px 1px;
Значения применяются к верху, правому краю, низу, и левому краю в соответствии с порядком исходного кода CSS	

Справочник сокращений

Граничные сокращения для различных свойств	Необходимо упомянуть еще, что можно даже задать значения свойств границ (border) только для одной границы элемента следующим образом: border-left-width: 2px; border-left-style: solid;

	border-left-color: black;
Сокращения для некоторых свойств полей (margin), заполнения (padding) и границы (border)	Все это действует таким же образом как было показано выше в разделе "Выбор между одиночным свойством и сокращенным значением".
Сокращения для шрифта	С помощью одной строки сокращения можно определить размер шрифта, толщину, стиль, семейство и высоту строки. Например, рассмотрим следующий код: font-size: 1.5em; line-height: 200%; font-weight: bold; font-style: italic; font-family: Georgia, "Times New Roman", serif Можно определить все это с помощью следующей строки: font: 1.5em/200% bold italic Georgia,"Times New Roman",serif;

Сокращение для фона	С помощью одной строки CSS можно определить цвет фона, фоновое изображение, повторение изображения и позицию изображения. Возьмем следующий код:
	
	background-color: #000;
	background-image: url(image.gif);
	background-repeat: no-repeat;
	background-position: top left;
	
	Это можно представить с помощью следующего сокращения:
	
	background:#000 url(image.gif) no-repeat top left;

Сокращения для списков	В данном случае аналогичная история со *свойствами списков* позволяет задать на одной строке значения свойств для типа маркера списка, позиции и изображения. Возьмем следующий код:
	
	list-style-type: circle;
	list-style-position: inside;
	list-style-image: url(bullet.gif);
	
	Это эквивалентно следующему:
	
	list-style: circle inside url(bullet.gif);

8.1.11 Применение CSS к HTML

Существует три способа применения CSS к документу HTML:

- строковые;
- вложенные;
- внешние таблицы стилей.

8.1.11.1 Строковые стили

Можно применить таблицу стилей к элементу, используя атрибут style следующим образом.

8.1.11.2 Вложенные стили

Вложенные таблицы стилей помещаются в заголовке документа внутри элемента style.

Преимущество вложенных таблиц стилей состоит в том, что не нужно добавлять атрибут style в каждый параграф - можно оформить их все с помощью одного единственного определения. Это означает также, что если потребуется изменить внешний вид всех параграфов, это можно будет сделать в одном единственном месте, однако все это, тем не менее, ограничено одним документом.

8.1.11.3 Внешние таблицы стилей

Внешние таблицы стилей означают размещение всех определений CSS в отдельном файле, сохраняя его с расширением файла CSS, и затем применение его к документам HTML, используя элемент link в заголовке документа.

Внешние таблицы стилей позволяют хранить все определения оформления в одном единственном файле. Это означает, что можно вносить изменения на всем сайте, изменяя только один файл. Веб-браузер при этом может загрузить его один раз и затем кешировать для всех других документов, которые на него ссылаются, что уменьшает объем загружаемой информации.

8.1.12 Импорт таблиц стилей

Существует и другой способ импорта внешних таблиц стилей в файлы HTML - оператор @import.

Двумя фундаментальными концепциями CSS являются наследование и каскадирование.

8.1.12.1 Наследование

Наследование в CSS является механизмом, с помощью которого определенные свойства передаются от элемента предка его элементам потомкам.

Используя наследование можно, например, определить свойства шрифта для элементов html или body, и они будут унаследованы всеми другими элементами. Можно определить цвета фона и переднего плана для определенного контейнерного элемента, и цвет переднего плана будет автоматически унаследован элементами потомками в этом контейнере.

Каждый элемент в документе HTML будет наследовать все наследуемые свойства своего предка, за исключением корневого элемента (html), который не имеет предка.

8.1.12.2 Каскадирование

Каскадирование - это механизм, который управляет конечным результатом, в случае когда несколько конфликтующих объявлений CSSприменяются к одному элементу. Имеется три основные концепции, которые управляют порядком, в котором применяются объявления CSS:

- Важность
- Специфичность
- Порядок исходного кода

Важность является наиболее значимой.

Важность объявления CSS зависит от того, где оно определено. Конфликтующие объявления будут применяться в следующем порядке, более поздние будут переопределять предыдущие:

- Таблицы стилей агента пользователя
- Обычные объявления в таблицах стиля пользователя
- Обычные объявления в таблицах стиля автора
- Важные объявления в таблицах стиля автора
- Важные объявления в таблицах стиля пользователя

Специфичность определяют как меру того, насколько конкретным является селектор некоторого правила. Селектор с низкой специфичностью может соответствовать многим элементам (такой как *, который соответствует каждому элементу в документе), в то время как селектор с высокой специфичностью может соответствовать только одному элементу на страницу (такой как #nav, который соответствует только элементу с id совпадающим с nav).

Специфичность имеет четыре компоненты, которые можно обозначить как a, b, c и d. Компонента a является наиболее разграничивающим, d - наименее.

- Компонента a определяется очень просто: это 1 для объявления атрибута style, иначе это 0.
- Компонента b является числом селекторов id в селекторе (тех, которые начинаются с#).
- Компонента c является числом селекторов атрибутов, включая селекторы классов - и псевдо-классов.
- Компонента d является числом типов элементов и псевдо-элементов в селекторе.

Если два объявления влияют на один и тот же элемент, имеют одинаковую важность и одинаковую специфичность, то окончательное решение определяет ***порядок исходного кода***. Объявление, которое появляется позже в таблицах стилей будет выигрывать у тех, которые встречаются раньше.

8.1.13 Спецификация CSS2

Эта спецификация определяет Каскадные таблицы Стилей, уровень 2 (CSS2). CSS2 - это язык таблиц стилей, позволяющий авторам и пользователям подключать стили (например, шрифты, пробелы и звуковые сигналы) в структурированные документы (например, документы HTML и приложения XML). CSS2 упрощает создание и обслуживание Web -сайта путем разделения структуры и стиля представления документов.

8.2 Контрольные вопросы

1. Какое отличие HTML от CSS?
2. Перечислите уровни CSS.
3. Назовите преимущества и недостатки использования CSS.
4. Для чего нужен селектор?
5. Для чего нужны универсальные селекторы?
6. Каково назначение сокращенной записи?
7. Перечислите способы применения CSS к документу HTML.
8. Назовите фундаментальные концепции CSS и дайте им определение.
9. Дайте определения понятиям: каскадные таблицы стилей, стиль, таблица стилей, каскадирование.

10.С какими проблемами сталкивались разработчики сайтов до появления CSS?
11.Как оформить комментарий в CSS?

8.3. Задания

1. Напишите код, используя базовые конструкции CSS. (R0801)
2. Напишите код, используя селекторы, объединенные в группы (R0802)
3. Напишите код с использованием селекторов атрибутов элементов (R0803)
4. Напишите код с использованием селекторов потомков элементов (R0804)
5. Напишите код, используя селекторы нижележащих элементов (R0805)
6. Напишите код, используя селекторы смежных одноуровневых элементов (R0806)
7. Создайте данную страницу по образцу с помощью HTML и CSS, используя селекторы. (R0808)