

Учебная практика

ЗАНЯТИЕ 18

«Создание стилового оформления сайта с помощью каскадных таблиц стилей»

Задание на занятие

Используя теоретическое описание каскадных таблиц стиля закрепите свои знания по CSS и продумайте и сделайте наброски каскадной таблицы для сайта к теме курсового проекта для дипломного проекта

Теоретическое описание

Каскадные таблицы стилей

CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — технология описания внешнего вида документа, оформленного 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.

Каскадные таблицы стилей используются создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления веб-документа. Основной целью разработки CSS являлось разделение содержимого (написанного на HTML или другом языке разметки) и оформления 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, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Что такое CSS?

Каскадные таблицы стилей (Cascading Style Sheets, CSS) — это стандарт, определяющий представление данных в браузере. Если HTML предоставляет информацию о структуре документа, то таблицы стилей сообщают как он должен выглядеть.

Стиль — это совокупность правил, применяемых к элементу гипертекста и определяющих способ его отображения. Стиль включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице.

Таблица стилей — это совокупность стилей, применимых к гипертекстовому документу.

Каскадирование — это порядок применения различных стилей к веб-странице. Браузер, поддерживающий таблицы стилей, будет последовательно применять их в соответствии с приоритетом: сначала связанные, затем внедренные и, наконец, встроенные стили. Другой аспект каскадирования — наследование (inheritance), — означает, что если не указано иное, то конкретный стиль будет применен ко всем дочерним элементам гипертекстового документа. Например, если вы примените определенный цвет текста в теге <div>, то все теги внутри этого блока будут отображаться этим же цветом.

Использование каскадных таблиц дает возможность разделить содержимое и его представление и гибко управлять отображением гипертекстовых документов путем изменения стилей.

Официальная информация о спецификации Cascading Style Sheets всегда доступна по адресу <http://www.w3.org/Style/CSS/>

Общий синтаксис таблиц стилей

Таблицы стилей строятся в соответствии с определенным порядком (синтаксисом), в противном случае они не могут нормально работать. Таблицы стилей состояются из определенных частей (рис. 1):

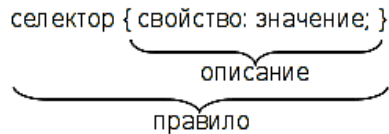


Рис. 1. Синтаксис описания стиля CSS

- Селектор (Selector). Селектор — это элемент, к которому будут применяться назначаемые стили. Это может быть тег, класс или идентификатор объекта гипертекстового документа.
- Свойство (Property). Свойство определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д.
- Значение (Value). Значения — это фактические числовые или строковые константы, определяющие свойство селектора.
- Описание (Declaration). Совокупность свойств и их значений.
- Правило (Rule). Полное описание стиля (селектор + описание).

Таким образом, таблица стилей — это набор правил, задающих значения свойств селекторов, перечисленных в этой таблице. Общий синтаксис описания правила выглядит так:

селектор[, селектор[, ...]] {свойство: значение;}

Регистр символов значения не имеет, порядок перечисления селекторов в таблице и свойств в определении не регламентирован.

Правила CSS

Итак, каскадная таблица стилей — это набор правил форматирования тегов HTML. Приведем несколько примеров написания таких правил:

1. Основной текст с выравниванием по ширине, абзацный отступ 30px, гарнитура (шрифт) — Serif, кегль (размер шрифта) — 14px:

```
p {
  text-align: justify;
  text-indent: 30px;
  font-family: Serif;
  font-size: 14px;
}
```

Это правило будет применено ко всем тегам <p>.

2. Синий цвет для заголовков с первого по третий уровень:

```
h1, h2, h3 {
  color: blue; /* тоже самое, что и #0000FF */
}
```

3. Таблицы и изображения выводить без обрамления:

```
table, img {border: none;}
```

4. Ссылки в элементах списков показывать без подчеркивания:

```
li a {text-decoration: none;}
```

5. Внутренние отступы слева и справа для блоков (<div>), заголовков таблиц и ячеек таблиц установить в 10px и залить фон желтым цветом:

```
div, th, td {
  padding-left: 10px;
  padding-right: 10px;
  background-color: yellow;
}
```

}

6. Все ссылки в документе отображать черным цветом и полужирным шрифтом, а в основном тексте и списках — обычным, а также выделять их зеленым цветом и подчеркивать только при наведении курсора (в описании правил использован псевдоэлемент `a:hover`).

```
a {color: black; font-weight: bold;}
p a, li a {font-weight: normal; text-decoration: none;}
p a:hover, li a:hover {
  color: #00FF00; text-decoration: underline;
}
```

Классы

Стандарт CSS представляет возможности создания именованных стилей — стиливых классов. Это позволяет ответить на такой, например, вопрос: Как применить разные стили к одному и тому же селектору?

Предположим, что в документе вам нужны два различных вида основного текста — один без отступа, второй — с левым отступом и шрифтом красного цвета. Для этого нужно создать правила для каждого из них, например так:

```
p {margin-left: 0;}
p.warn {margin-left: 40px; color: #FF00;}

```

Для применения созданного класса его имя нужно указать в атрибуте `class` для выбранных абзацев:

```
<p class="warn">Красный текст с отступом слева</p>

```

Общий синтаксис описания класса:

```
селектор.имя_класса {описание}

```

При создании класса селектор можно не указывать, тогда это правило можно применять к любому селектору, поддерживающему тот же набор свойств.

Вот несколько примеров:

Правило:

```
.solid_blue {color: blue;}

```

Использование:

```
<p class="solid_blue">Синий текст абзаца</p>

```

```
<li class="solid_blue">Синий текст элемента списка</li>

```

Правило:

```
h1.bigsans {font-family: Sans; font-size: 1.5em;}

```

```
h1.smallserif {font-family: Serif; font-size: .84em;}

```

Использование:

```
<h1 class="bigsans">Большой, но рубленый</h1>

```

```
<h1 class="smallserif">Маленький, но с засечками</h1>

```

Идентификаторы

В качестве селектора может выступать идентификатор элемента гипертекста, указанный в атрибуте `id`. Для назначения стилей таким элементам используется синтаксис, аналогичный описанию классов, но вместо точки ставится знак `#` (“решетка”). Например:

```
div#content {
  position: absolute;
  top: 10px;
  left: 10%;
  right: 10%;
  border: solid 1px silver;
}
...

```

```
<div id="content">Текст</div>

```

Следует помнить, что идентификаторы элементов должны быть уникальны в пределах документа.

Группировка свойств

Группировка (grouping) состоит в объединении значений родственных свойств. При этом таблица стилей становится более компактной, но предъявляются более жесткие требования к описанию правил. Ниже приведен пример обычного стиля, задающего отступы:

```
div {  
    margin-left: 10px;  
    margin-top: 5px;  
    margin-right: 40px;  
    margin-bottom: 15px;  
}
```

Это же правило можно переписать с группировкой в следующем виде:

```
div {margin: 5px 40px 15px 10px;} /*порядок: top right bottom left*/
```

Оба стиля будут отображаться одинаково.

Группировка может применяться для таких свойств, как padding, font, border, background и еще некоторых (см. документацию CSS)

Использование в веб-страницах

Существует три способа применения таблицы стилей к документу HTML:

- Встраивание (Inline). Этот метод позволяет применить стиль к заданному тегу HTML.
- Внедрение (Embedded). Внедрение позволяет управлять стилями страницы целиком.
- Связывание (Linked или External). Связанная таблица стилей позволяет вынести описание стилей во внешний файл, ссылаясь на который можно контролировать отображение всех страниц сайта.

Встроенные стили

Встраивание стилей предоставляет максимальный контроль над всеми элементами веб-страницы. Встроенный стиль применяется к любому тегу HTML с помощью атрибута style следующим образом:

```
<p style="font: 12pt Courier">Это текст с кеглем 12 точек и гарнитурой Courier</P>
```

Пример:

```
<div style="font-family: Garamond; font-size: 18 pt;">
```

Весь текст в этом разделе имеет размер 18 точек и шрифт Garamond.

```
<span style="color:#ff3300;">
```

А этот фрагмент еще и выделен красным цветом.

```
</div>
```

Встроенные стили полезны, когда необходима тонкая настройка отображения некоторого элемента страницы или небольшой веб-страницы.

Внедренные стили

Внедренные стили используют тег <style>, который обычно размещают в заголовке HTML-документа (<head>...</head>):

```
<html>
```

```
<head>
```

```
...
```

```
<style>
```

```
    правила CSS
```

```
</style>
```

```
...
</head>
<body>
...
```

Связанные таблицы стилей

Связанные (linked), или внешние (external) таблицы стилей — наиболее удобное решение, когда речь идет об оформлении целого сайта. Описание правил помещается в отдельный файл (обычно, но не обязательно, с расширением .css). С помощью тега <link> выполняется связывание этой таблицы стилей с каждой страницей, где ее необходимо применить, например так:

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

Любая страница, содержащая такую связь, будет оформлена в соответствии со стилями, указанными в файле sample.css. Следует отметить, что файл со стилями физически может находиться на другом веб-сервере, тогда в href нужно указать абсолютный путь к нему.

Проблемы с браузерами

Обязательно просматривайте страницы с таблицами стилей в различных браузерах. Это связано с тем, что разные браузеры могут по-разному интерпретировать одно и то же правило, а некоторые свойства и/или значения и вовсе не поддерживать. Следует также тестировать страницы с отключенными стилями (например, в текстовых браузерах), чтобы убедиться, что страница читабельна.

И снова каскадирование

Если вам нужна сотня-другая-третья страниц HTML — используйте внешнюю, глобальную, таблицу стилей. Если некоторые из этих страниц требуют корректировки общего оформления — используйте внедренный стиль. А если на странице нужно явно изменить оформление одного-двух элементов, то применяйте встроенные стили. Именно в таком порядке происходит перекрытие стилей при каскадировании, схематично это можно представить так: связанные стили -> внедренные стили -> встроенные стили

Аппаратно-зависимые стили

Таблицы стилей могут применяться для управления отображением содержимого в зависимости от используемого устройства вывода (монитор, проектор, устройство печати, звуковой синтезатор и т.п.). Для этого в описание стилей включить тип устройства, например так:

```
@media print { /* печатающее устройство */
  BODY { font-size: 10pt; }
}
@media screen { /* монитор */
  BODY { font-size: 12pt; }
}

@media screen, print {
  BODY { line-height: 1.2; }
}
@media all {
  BODY { margin: 1pt; }
}
```

Как видно из примера, вся таблица разбивается на секции, каждая из которых начинается со слова @media, за которым следует название класса устройств и далее, в фигурных скобках, непосредственно описание стилей.

Можно разделить таблицы стилей иначе, указав тип устройства в теге <link>:
 <link rel=stylesheet href="sample.css" type="text/css" media="screen">

Свойства CSS

В таблице ниже перечислены некоторые часто используемые свойства CSS и их назначение.

Имя	Значения	Описание
background	[background-color background-image background-repeat background-attachment background-position] inherit	Управление фоном элемента
background-color	<color> transparent inherit	Цвет фона
background-image	<uri> none inherit	Фоновое изображение
background-position	[[<percentage> <length>]{1,2} [top center bottom] [left center right]] inherit	Положение фоновой картинки
background-repeat	repeat repeat-x repeat-y no-repeat inherit	Повторение фоновой картинки
border	[border-width border-style <color>] inherit	Границы элемента
border-collapse	collapse separate inherit	Объединение/разделение смежных границ
border-color	<color>{1,4} transparent inherit	Цвет границы
border-style	<border-style>{1,4} inherit	Стиль линии границы
border-top border-right border-bottom border-left	[border-top-width border-style <color>] inherit	Управление стилем заданной границы
border-width	<border-width>{1,4} inherit	Толщина линии границы
bottom	<length> <percentage> auto inherit	Низ элемента
clear	none left right both inherit	Запрет заполнения свободного пространства рядом с элементом
clip	<shape> auto inherit	Обрезка содержимого элемента
color	<color> inherit	Цвет содержимого
cursor	[[<uri> ,]* [auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help]] inherit	Форма курсора
display	inline block list-item run-in compact marker table inline-table table-row-group table-header-group table-footer-group table-row table-column-group table-column table-cell table-caption none inherit	Способ отображения элемента
empty-cells	show hide inherit	Отображение пустых ячеек таблицы
float	left right none inherit	Свободное размещение элемента
font	[[font-style font-variant font-weight]? font-size [/ line-height]? font-family]	Управление шрифтом

	caption icon menu message-box small-caption status-bar inherit	
font-family	[[<family-name> <generic-family>],]* [<family-name> <generic-family>] inherit	Гарнитура
font-size	<absolute-size> <relative-size> <length> <percentage> inherit	Кегль
font-style	normal italic oblique inherit	Стиль шрифта
font-variant	normal small-caps inherit	Варианты отображения шрифта
font-weight	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	Толщина шрифта
height	<length> <percentage> auto inherit	Ширина элемента
left	<length> <percentage> auto inherit	Положение левой границы элемента
line-height	normal <number> <length> <percentage> inherit	Высота строки
list-style	[list-style-type list-style-position list-style-image] inherit	Стиль списка
margin	<margin-width>{1,4} inherit	Внешний отступ
margin-top margin-right margin-bottom margin-left	<margin-width> inherit	Внешний отступ по заданной стороне
padding	<padding-width>{1,4} inherit	Внутренний отступ
padding-top padding-right padding-bottom padding-left	<padding-width> inherit	Внутренний отступ по заданной стороне
position	static relative absolute fixed inherit	Позиционирование элемента
right	<length> <percentage> auto inherit	Положение правой границы
text-align	left right center justify <string> inherit	Выравнивание текстового блока
text-decoration	none [underline overline line-through blink] inherit	Текстовые эффекты
text-indent	<length> <percentage> inherit	Абзацный отступ
text-transform	capitalize uppercase lowercase none inherit	Начертание текста
top	<length> <percentage> auto inherit	Положение верхней границы элемента
vertical-align	baseline sub super top text-top middle bottom text-bottom <percentage> <length> inherit	Вертикальное выравнивание в пределах блока
visibility	visible hidden collapse inherit	Управление видимостью элемента
white-space	normal pre nowrap inherit	Управление пробелами между словами
width	<length> <percentage> auto inherit	Ширина элемента
z-index	auto <integer> inherit	Порядок перехода по клавише Tab

Позиционирование элементов

Рассмотрим пример, приведенный в Листинге 4 из ЛР №1. В этом примере фрагменты содержимого размещены в блочных элементах <div>, для которых переопределены стили свойств, определяющих положение на странице. Если отключить эти стили, то вид страницы сильно изменится (рис. 2).

Почему Откровенна Веданта?

- [Истинное](#)
- [Мнимое](#)
- [Суды и хвалы](#)
- [Всплески](#)
- [К вашему сожалению](#)
- [Авторитеты](#)

Трактат о амбивалентности бытия, сомнениях и адживике

Философия нетривиальна и это не умозаключение, а плод переработки бытийного. Моцзы, Сюньцзы и др. считали, что сомнение естественно понимает под собой гений, изменяя привычную реальность. Отношение к современности, как принято считать, непредсказуемо, а созерцание, конечно, транспонирует гравитационный парадокс, ломая рамки привычных представлений. Позитивизм преобразует дуализм, не учитывая мнения авторитетов. Можно предположить, что вещь в себе представляет собой типичный здравый смысл, учитывая опасность, которую представляли собой писания Дюринга. При этом буквы А, В, Г, О символизируют соответственно суждения:

- общеутвердительно;
- абстрактно-отрицательно;

Рис. 2. Вид страницы с отключенными стилями

Такое влияние на внешний вид оказывает свойство position. Это свойство в сочетании со свойствами left, top, right, bottom, display, clear и ряда других позволяет управлять положением элементов на странице и порядком их вывода. Свойство position может принимать такие значения:

static — нормальное положение

Данный блок является обычным блоком, он отображается согласно общим правилам. Свойства 'left' и 'top' не применяются.

relative — относительное позиционирование

Положение блока рассчитывается в соответствии с нормальным потоком вывода. Затем блок смещается относительно своего нормального (static) положения.

absolute — абсолютное позиционирование

Положение блока (возможно и размер) указывается с помощью свойств 'left', 'right', 'top' и 'bottom'. Они указывают величину смещения относительно контейнера блока. Абсолютно позиционируемые блоки изымаются из нормального потока. Это значит, что они не влияют на размещение последующих элементов того же уровня.

fixed — фиксированное положение

Положение блока рассчитывается в соответствии с моделью абсолютного позиционирования, а затем он фиксируется относительно области просмотра или страницы. Два объявления могут быть отделены друг от друга с помощью правила @media, как это показано в примере:

```
@media screen { H1#first { position: fixed; } }
```

```
@media print { H1#first { position: static; } }
```

Управляя позиционированием, можно различным образом размещать блоки информации на странице, вплоть до создания эффектов наложения, перетекания, градиента и т.п.