

Internet- технологии

ЛЕКЦИЯ №4
КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ

Содержание

- Основные понятия CSS.
- Способы подключения каскадных таблиц стилей.
- Селекторы CSS.

- В следующей лекции:
 - Способы позиционирования элементов. Плавающая модель.
 - Браузер-специфические свойства.
 - CSS-препроцессоры.



Назначение каскадных таблиц стилей

CSS (Cascading Style Sheets, каскадные таблицы стилей) – это язык стилей, определяющий внешний вид документа, написанного с использованием языка разметки. CSS используется при создании веб-страниц для задания цвета, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида веб-документов.

Преимущества CSS:

- позволяют разделить смысловое содержимое страницы и оформление;
- управление отображением множества документов с помощью одной таблицы стилей;
- более точный контроль над внешним видом страниц;
- различные представления для разных носителей информации (экран, печать, и т. д.);
- сложная и проработанная техника дизайна.



Синтаксис CSS

```
selector {property: value;}
```

↑
selector
/селектор:
К какому
HTML-тэгу
(тэгам)
применяется
свойство
(например,
"body")

↑
property
/свойство:
Свойство, которое, к примеру,
может быть цветом фона
("background-color")

↙
value/значение:
Значением свойства
background-color
может быть, например
red ("#FF0000")

```
body {  
    background-color: #FF0000;  
}
```



Синтаксис CSS. Определение стиля

селектор {свойство:значение;}

- Если **значение** состоит из нескольких слов, то оно помещается в одинарные или двойные **кавычки**. Внутри строки можно комбинировать типы кавычек или добавить перед кавычкой слэш.
- Если необходимо перечислить **несколько значений** свойства, то они разделяются **пробелом**.
- **Альтернативные значения** разделяются **запятой**.
- Если необходимо определить **более одного свойства**, то они разделяются **точкой с запятой**.
- Селекторы, для которых определяют **одинаковые свойства**, допускается **группировать**. Селекторы в группе разделяются **запятыми**.

```
h1, h2, h3 {  
    font-family: 'Times New Roman', serif;  
    font-size: 14pt; /* размер текста */  
}
```



Синтаксис CSS. Пример

Каскадность таблиц заключается возможности наследования вложенными элементами стилей элементов-контейнеров. Кроме того, стили, подключенные разными способами, могут переопределяться.

Описание стилей:

```
body {  
    font-family: Arial, Verdana, sans-serif;  
    font-size: 11pt;  
    background-color: #fofofo;  
    color: #333;  
}  
h1 {  
    color: #a52a2a;  
    font-size: 24pt;  
    font-family: Georgia, Times, serif;  
    font-weight: normal;  
}
```

Тело документа:

```
<body>  
  <h1> Introduction</h1>  
  <p> A single CSS file can describe a common  
  "style" to be applied to many HTML, XHTML, and  
  XML documents.</p>  
</body>
```

Introduction

A single CSS file can describe a common "style" to be applied to many HTML, XHTML, and XML documents.



Единицы измерения в CSS

Меры длины, используемые при указании значений свойств для указания размеров:

- 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** – points, точка (1pt = 1/72in)), пиках (**pc** – picas (1pc = 12pt)), пикселях (**px**), 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**)



Задание цвета в CSS

Цвет для тех свойств, где это нужно, может быть определен одним из следующих способов:

- при помощи **названия** цвета: yellow, red, green, grey, ..
- шестнадцатеричным заданием цвета в формате **#rrggbb**: #ff0000, #883490, #ffffff, ..
- шестнадцатеричным заданием цвета в формате **#rgb** (например, #0c0)
- десятичным заданием составляющих цвета в формате **rgb(x,x,x)**, где "x" число от 0 до 255 (rgb(red, green, blue) – например, rgb(255,0,0), rgb(100,23,78), ..)
- **rgb(x%,x%,x%)** – где "x%" число от 0.0 до 100.0 (например, 0%,80%,0%)



Подключение таблиц стилей

Существуют следующие способы использования стилей:

- **inline-описание** – используя атрибут `style`, в котором можно задать значения свойств элемента в соответствии с синтаксисом каскадных таблиц стилей;
- описание в **секции заголовка** в стилевом блоке, ограниченном тэгами `<style>` и `</style>`;
- описать во **внешнем файле**, подключаемом к веб-документу при помощи тэга `<link>`, задаваемого в разделе `<head>`;

Одно из проявлений **каскадности** таблиц стилей заключается в том, что стили, подключенные разными способами, могут переопределять друг друга. Приведенный выше список способов внедрения стилей соответствует порядку переопределения. Вышерасположенный способ подключения может переопределять нижерасположенный.



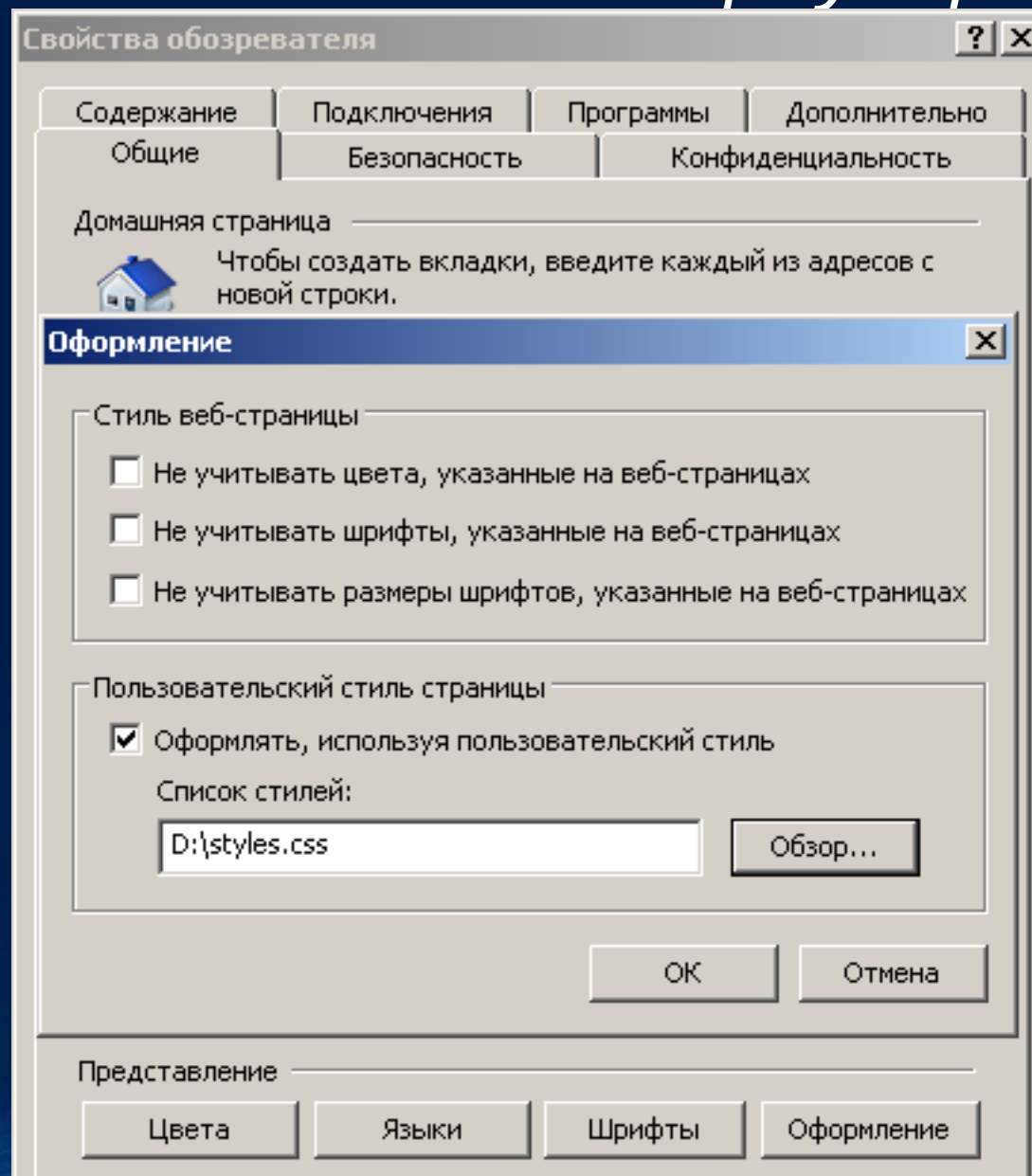
Приоритет стилей

Приоритеты использования стилей, по возрастанию:

- Наиболее низким приоритетом обладает **стиль браузера**;
- Следующим по значимости является стиль, **заданный пользователем** браузера в его настройках;
- Наивысшим приоритетом обладает стиль, заданный **автором страницы**:
 - Самым низким приоритетом обладают стили, **наследуемые** в документе элементом от своих предков;
 - Более высоким приоритетом обладают стили, заданные **во внешних таблицах стилей**, подключённых к документу;
 - Ещё более высоким приоритетом обладают стили, **содержащиеся в контейнерах style** данного документа.
 - Ещё более высоким приоритетом обладают стили, объявленные непосредственно в теге данного элемента посредством **атрибута style**;
 - И наконец самым высоким приоритетом обладают стили, объявленные автором страницы с помощью сопроводительного слова **!important**.



Указание стиля браузера



Подключение таблиц стилей. inline-описание

inline-описание – разновидность внутренних таблиц стилей.

Правила оформления могут располагаться в веб-документе в теле отдельного тега (посредством его атрибута **style**). Описанные правила будут действовать только на содержимое этого тега.

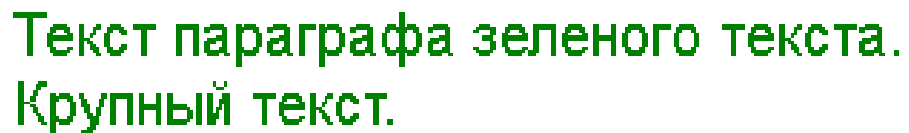
```
<body>
```

```
  <p style="font-size: 20px; color: green; font-family: arial, helvetica,  
  sans-serif">
```

Текст параграфа зеленого текста. Крупный текст.

```
  </p>
```

```
</body>
```



Текст параграфа зеленого текста.
Крупный текст.



Подключение таблиц стилей в секции заголовка

Таблица стилей может быть описана в самом документе, располагаясь в нём между тегами <style> и </style> (которые, в свою очередь, располагаются в этом документе между тегами <head> и </head>). Все правила этой таблицы действуют на протяжении всего документа.

```
<head>
  <title>Example</title>
  <style type="text/css">
    body {
      color: red;
    }
  </style>
</head>
<body>
  <p>This is a red page</p>
</body>
```

This is a red page



Подключение таблиц стилей. Внешний файл

Таблицы стилей могут располагаться в отдельных файлах, имеющих формат CSS (это обычный текстовый файл, в котором не содержится ничего, кроме перечня правил CSS и комментариев к ним). Подключение выполняется посредством тега `<link>` секции заголовка, имеющего атрибут `href`, со значением адреса подключаемой таблицы стилей. Таким образом можно подключать таблицу стилей, которая находится на другом сайте. Все правила этой таблицы действуют на протяжении всего документа.

```
<head>
  <title>Example</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Заголовок</h1>
  <p>Текст</p>
</body>
```

```
h1 {
  color: #000080;
  font-size: 200%;
  text-align: center;
}
p {
  padding-left: 20px;
}
```



Подключение таблиц стилей. Внешний файл

Подключение к веб-документу возможно посредством директивы @import, располагающейся в этом документе между тегами <style> и </style>, которая также указывает на адрес таблицы стилей. Правило @import позволяет импортировать содержимое CSS-файла в текущую стилевую таблицу.

Подключение к веб-документу:

```
<style media="screen">  
    @import url(styles.css);  
</style>
```



Импорт в состав стилевого файла:

```
@import "style/print.css";  
@import "style/palm.css";  
h1 {  
    color: #000080;  
    font-size: 200%;  
    text-align: center;  
}
```



Селекторы

Селектор определяет элементы, для которых применяются стили – набор свойств и их значений.

Комбинации
селекторов:

- контекстные селекторы.
- дочерние селекторы.
- соседние селекторы.
- родственные селекторы.

Вид селектора	Пример	Пример стиля
универсальный	*	* { color:green; }
селектор тегов	div	div { margin:7px; padding:7px; }
селектор атрибута	q[title]	q[title] { color: maroon; }
селектор класса	.important	.important { font-weight:bold; color:red; }
селектор id	#onlyThisOne	#onlyThisOne { font-family:courier; }
псевдокласс	a:link	a:link { color:blue; }
псевдоэлемент	p:first-letter	p:first-letter { color:red }



Универсальный селектор

Универсальный селектор соответствует любому элементу веб-страницы, позволяет определить один стиль для всех – например, задать шрифт или начертание текста.

* { Описание правил стиля }

Записи *.class и .class являются идентичными по своему результату.

```
<head> <style>
  * { font-style: italic; font-size: 96%; }
</style> </head>
<body>
  <br> /* Пример кода стилевого файла */
<pre> body {
  background-image: url(background.png);
  background-position: left top;
          background-repeat: repeat-x;
} </pre>
</body>
```

/ Пример кода стилевого файла */*

```
body {
  background-image: url(background.png);
  background-position: left top;
  background-repeat: repeat-x;
}
```



Селекторы тегов

В качестве селектора может выступать любой элемент HTML, для которого определяются правила форматирования, такие как: цвет, фон, размер и др.

```
<head>
```

```
<style>
```

```
  a {
```

```
    text-decoration: none;
```

```
    color: red;
```

```
  }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
  <a href="http://htmlbook.ru"> http://htmlbook.ru</a> (оффлайн-версии <a  
href="Additional/Справочник%20CSS.chm"> справочника</a>, 2011 г) и <a  
href="Additional/Самоучитель%20CSS.chm"> самоучителя</a>)
```

```
</body>
```

<http://htmlbook.ru> (оффлайн-версии
справочника, 2011 г) и самоучителя)



Селекторы атрибутов

Использование селектора атрибута (общий синтаксис – Селектор[атрибут] { Описание правил стиля }, пробел между именем селектора и квадратными скобками не допускается) позволяют установить стиль для элементов, для которых определён атрибут (или по его значению).

```
<style>
  a[target="_blank"] {
    background: url(image/blank.png) 0 6px no-repeat;
    padding-left: 15px;
  }
</style>
<body>
  <p><a href="1.html"> Обычная ссылка </a>
    <a href="2.html" target="_blank"> Ссылка в новом окне </a></p>
</body>
```

Обычная ссылка

 Ссылка в новом окне



Селекторы атрибутов

Селектор атрибутов позволяет сократить количество ID и классов и сделать верстку более легкой для поддержки. Помимо простого селектора атрибутов, в CSS доступны следующие разновидности:

[att^="value"] – выбор элементов по атрибуту, который начинается с заданного значения.

[att|="value"] – выбор элементов по атрибуту, значение которого либо совпадает с value, либо начинается с value, за которым следует дефис (-)

[att\$="value"] – по атрибуту, который заканчивается заданным значением.

[att*="value"] – по атрибуту, в котором содержится заданное значение.

[att ~="value"] – по атрибуту, у которого среди перечисленных значений есть совпадения с требуемым.

```
a[title$="ссылка"] { display: block; }  
[class~="block"] h3 { color: green; }
```

```
<a title="Это ссылка">Link</a>  
<div class="block tag">  
  <h3> Заголовок</h3>  
</div>
```



Селекторы. Классы

Для того, чтобы отнести элемент разметки к тому или иному классу, нужно воспользоваться его атрибутом `class`. Имя класса не является каким-либо стандартным именем элемента HTML-разметки. Оно определяет описание класса элементов разметки, которые будут отображаться одинаково.

Имена классов должны начинаться с латинского символа и могут содержать в себе символ дефиса (-) и подчеркивания (_). Использование русских букв в именах классов недопустимо.

```
.important {  
  font-weight:bold; color:red;  
}
```

```
<p class="important"> Important Text</p>  
<p> Ordinary Text </p>  
<div class="important"> Important </div>
```

Important Text

Ordinary Text

Important



Селекторы. Классы. Пример

```
<head>
  <style>
    div {
      writing-mode:tb-rl; /* Отдельные стили для IE */
      font-size: 196%;
    }
    .vertical {
      width: 1em;   line-height: 1em;
    }
  </style>
</head>
<body>
  <div style="text-indent:0px" class="vertical">
В е р т и к а л ь </div>
</body>
```

В
е
р
т
и
к
а
л
ь



Селекторы. Идентификаторы

Селекторы id (идентификаторы, #<Имя идентификатора> { Описание правил стиля }), применяются для элементов с атрибутом id и соответствующим его значением. Основное отличие классов от идентификаторов состоит в том, что имена вторых должны быть уникальными, не повторяться, что позволяет их использовать вместе с JavaScript.

```
div#First {  
    background-color: silver;  
}
```

```
<div id="First">Идентификаторы  
широко используются в JavaScript для  
нахождения уникального элемента в  
документе.</div>
```

Идентификаторы широко используются в JavaScript для нахождения уникального элемента в документе.



Селекторы. Идентификаторы. Примеры

```
#rectangle {  
  width: 200px;  
  height: 100px;  
  background: red;  
}
```

```
#triangle-up {  
  width: 0;  
  height: 0;  
  border-left: 50px solid transparent;  
  border-right: 50px solid transparent;  
  border-bottom: 100px solid red;  
}
```

```
#triangle-left {  
  width: 0;  
  height: 0;  
  border-top: 50px solid transparent;  
  border-right: 100px solid red;  
  border-bottom: 50px solid transparent;  
}
```

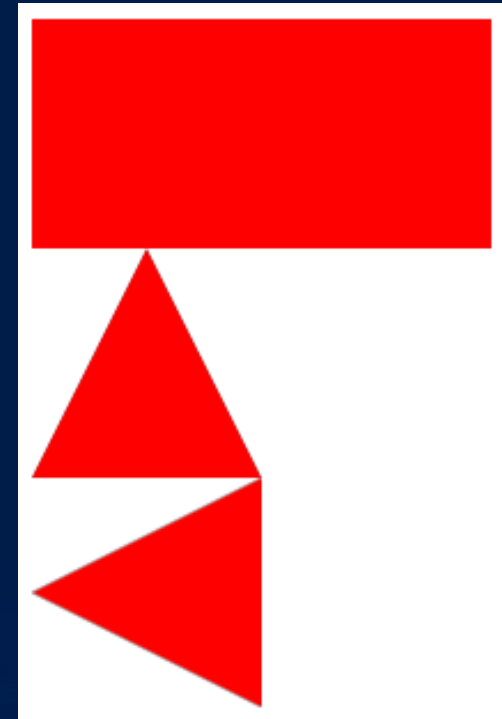
```
<body>
```

```
<div id="rectangle"></div>
```

```
<div id="triangle-up"></div>
```

```
<div id="triangle-left"></div>
```

```
</body>
```



Селекторы. Псевдоклассы

Псевдо-классы предназначены для изменения стиля существующих элементов страницы в зависимости от их динамического состояния, например при работе со ссылками (:link, :visited, :hover, :active, :focus).

- link

```
a:link { color: #6699CC; }
```

Это - непосещённая ссылка

Это - непосещённая ссылка

- visited

```
a:visited { color: #660099; }
```

Это - посещённая ссылка

Это - посещённая ссылка

- active

```
a:active { background-color: #FFFF00; }
```

Нажмите здесь и держите клавишу нажатой

Нажмите здесь и держите клавишу нажатой

- hover

```
a:hover { color: orange; font-style: italic; }
```

Проведите указателем мыши над этой ссылкой

Проведите указателем мыши над этой ссылкой



Селекторы. Псевдоклассы

:checked – применяется к элементам интерфейса, таким как переключатели (checkbox) и флажки (radio), когда они находятся в положение «включено».

:indeterminate – задаёт стиль для элементов форм, таким как флажки и переключатели, когда они находятся в неопределённом состоянии.

:disabled – применяет стиль к заблокированным элементам форм.

:enabled – используется для применения стиля к доступным (не заблокированным) элементам форм.

:optional – применяет стилевые правила к полю формы, у которого не задан атрибут required.

:required – применяет стилевые правила к элементу input, у которого установлен атрибут required.

:invalid – применяется к полям формы, содержимое которых не соответствует указанному типу.



Селекторы. Псевдоклассы

:empty – представляет пустые элементы, т. е. те, которые не содержат дочерних элементов, текста или пробелов.

:not – задаёт правила стилей для элементов, которые не содержат указанный селектор.

:first-of-type – задаёт правила стилей для первого элемента в списке дочерних элементов своего родителя.

:nth-of-type – используется для добавления стиля к элементам указанного типа на основе нумерации в дереве элементов.

:nth-last-child – используется для добавления стиля к элементам на основе нумерации в дереве элементов.

:first-child – применяет стилевое оформление к первому дочернему элементу своего родителя.

:only-child – применяется к дочерним элементам, только если он единственный у родителя.



Селекторы. Псевдоэлементы

Псевдоэлементы позволяют задать стиль элементов не определённых в дереве элементов, а также **генерировать содержимое**, которого нет в исходном коде текста. В CSS3 псевдоэлементы обозначаются с двумя двоеточиями, чтобы синтаксис отличался от псевдоклассов.

Синтаксис:

селектор:псевдоэлемент { свойства }

селектор.класс:псевдоэлемент { свойства }

```
<head> <style>
  p::first-letter {
    font-family: "Times New Roman", Times, serif;
    font-size: 2em; color: red;
  }
</style> </head>
<body><p> Виды селекторов </p></body>
```

Виды селекторов



Вопросы

- Для чего нужны таблицы стилей?
- Каким образом осуществляется подключение таблиц стилей?
- В чем заключается каскадность стилей?
- Что такое селекторы?
- В чем разница использования селекторов класса и идентификатора.

