

Верстка web-страниц

Стили

Титова Ольга Ивановна
Минск, 2016



Содержание

1. Знакомство с CSS
2. Преимущество в использовании CSS
3. Синтаксис оформления стилевых параметров
4. Способы задания стилевого оформления
5. Подключение CSS к HTML-документу



CSS (англ. Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.



Назначение CSS - установить внешний вид какого-либо элемента (фрагмента) веб-страницы;

Таким образом, это **правило**, которое сообщает браузеру, что и каким образом форматировать



Преимущества

Разграничение кода и оформления

Идея о том, чтобы код HTML был свободен от элементов оформления вроде установки цвета, размера шрифта и других параметров, стара как мир. В идеале, веб-страница должна содержать только теги логического форматирования, а вид элементов задаётся через стили. При подобном разделении работа над дизайном и версткой сайта может вестись параллельно.



Преимущества

Разное оформление для разных устройств

С помощью стилей можно определить вид веб-страницы для разных устройств вывода: монитора, принтера, смартфона, планшета и др. Например, на экране монитора отображать страницу в одном оформлении, а при её печати — в другом. Эта возможность также позволяет скрывать или показывать некоторые элементы документа при отображении на разных устройствах.



Преимущества

Расширенные по сравнению с HTML способы оформления элементов

В отличие от HTML стили имеют гораздо больше возможностей по оформлению элементов веб-страниц. Простыми средствами можно изменить цвет фона элемента, добавить рамку, установить шрифт, определить размеры, положение и многое другое.



Преимущества

Ускорение загрузки сайта

При хранении стилей в отдельном файле, он кэшируется и при повторном обращении к нему извлекается из кэша браузера. За счёт кэширования и того, что стили хранятся в отдельном файле, уменьшается код веб-страниц и снижается время загрузки документов.

Кэшем называется специальное место на локальном компьютере пользователя, куда браузер сохраняет файлы при первом обращении к сайту. При следующем обращении к сайту эти файлы уже не скачиваются по сети, а берутся с локального диска. Такой подход позволяет существенно повысить скорость загрузки веб-страниц.



Преимущества

Единое стилевое оформление множества документов

Сайт это не просто набор связанных между собой документов, но и одинаковое расположение основных блоков, и их вид. Применение единообразного оформления заголовков, основного текста и других элементов создает преемственность между страницами и облегчает пользователям работу с сайтом и его восприятие в целом. Разработчикам же использование стилей существенно упрощает проектирование дизайна.



Преимущества

Централизованное хранение

Стили, как правило, хранятся в одном или нескольких специальных файлах, ссылка на которые указывается во всех документах сайта. Благодаря этому удобно править стиль в одном месте, при этом оформление элементов автоматически меняется на всех страницах, которые связаны с указанным файлом. Вместо того чтобы модифицировать десятки HTML-файлов, достаточно отредактировать один файл со стилем и оформление нужных документов сразу же поменяется.



CSS

Фактически определение стиля состоит из двух основных элементов:

селектор – указатель на объект, который подлежит форматированию (например, тег `p` – абзац)

блок объявления – формирующие команды (свойство и его значение)

```
селектор      свойство      значение
┌──────────┐  ┌──────────┐  ┌──────────┐
body { background: #ffc910; }
```



Селектор

Сообщает браузеру, к какому элементу применяется стиль

В роли селекторов могут выступать различные объекты и их комбинации

Пример

```
p {color: red;}
```

В роли селектора – тег p

```
div p {color: #653302;}
```

В роли селектора – теги p, находящиеся внутри тега div



Блок объявления стиля

Код, расположенный сразу за селектором, содержит все форматирющие команды, которые нужно применить к данному селектору; блок **начинается с открывающей и заканчивается закрывающей фигурной скобкой**

Пример

```
p {color: red;}
```



Объявление свойства

Между скобками блока объявления можно добавить одно или несколько определений или форматизирующих команд; каждое объявление имеет две части – **свойство и значение**;

Двоеточие отделяет имя свойства от его значения;

Любое объявление заканчивается **точкой с запятой**

Пример

```
p {color: red; font-size: 20px;}
```



Свойство

Имеется достаточно большой перечень команд форматирования объектов, называемых свойствами

Пример

```
p {color: red; font-size: 20px;}
```



Значение

Для каждого используемого свойства есть допустимое множество значений, которые можно применять

Пример

```
p {  
    color: red;  
    font-size: 20px;  
}
```



Правила применения

Расширенная форма записи

```
td { background: olive; }
```

```
td { color: white; }
```

```
td { border: 1px solid black; }
```

Компактная форма записи

```
td {
```

```
background: olive;
```

```
color: white;
```

```
border: 1px solid black;
```

```
}
```



Правила применения

Имеет приоритет значение, указанное в коде ниже

Если для селектора вначале задаётся свойство с одним значением, а затем то же свойство, но уже с другим значением, то применяться будет то значение, которое в коде установлено ниже

Разные значения у одного свойства

```
p { color: green; } p { color: red; }
```

В данном примере для селектора **p** цвет текста вначале установлен зелёным, а затем красным. Поскольку значение `red` расположено ниже, то оно в итоге и будет применяться к тексту.

! Избегаем подобных противоречий



Правила применения

Значения

У каждого свойства **может быть только соответствующее его функции значение**. Например, для `color`, который устанавливает цвет текста, в качестве значений недопустимо использовать числа.



Правила применения

Комментарии нужны, чтобы делать **пояснения** по поводу использования того или иного стилевого свойства, выделять разделы или писать свои заметки.

Комментарии **позволяют легко вспоминать логику и структуру селекторов, и повышают разборчивость кода**. Вместе с тем, добавление текста увеличивает объём документов, что отрицательно сказывается на времени их загрузки. Поэтому комментарии обычно применяют в отладочных или учебных целях, а при выкладывании сайта в сеть их стирают или существенно сокращают.

Чтобы пометить, что текст является комментарием, применяют следующую конструкцию `/* ... */`



Правила применения

```
/* Стиль сделан для ознакомительных целей */  
div { width: 200px; /* Ширина блока */  
margin: 10px; /* Поля вокруг элемента */  
float: left; /* Обтекание по правому краю */  
}
```

Как следует из данного примера, комментарии можно добавлять в любое место CSS-документа, а также писать текст комментария в несколько строк.

Вложенные комментарии недопустимы.



Таблицы стилей могут быть:

- **внутренние;**
- **внешние.**

В зависимости от того, где определена стилевая информация: непосредственно в самой веб-странице или в отдельном файле, который связан с веб-страницей



CSS может быть:

- вынесен в отдельный файл (рекомендуется);
- оставлен в HTML-документе, обрамленный парным тегом `<style>...</style>` в служебном блоке `<head>`
- прописан в самом теге, к которому применяют стилевое оформление



Внешние таблицы стилей.

Это не что иное как **текстовый файл**, содержащий весь необходимый набор стилей CSS

Он **не должен включать в себя html-код**, поэтому не нужно добавлять сюда тег `<style>`

Имя данного файла всегда должно заканчиваться **расширением .css**



При вынесении CSS в отдельный файл:

- В самом html-коде указываем ссылку на внешний файл, где хранятся стили:

```
<html>  
  <head>  
    <title> ... </title>  
    <link rel="stylesheet" href="style.css" />  
  </head>  
  <body> ...
```

- В отдельном файле **style.css** прописываются стили (сам файл создаем в том же редакторе, но при сохранении задаем расширение *.css)



```
<link rel="stylesheet" href="style.css">
```

rel="stylesheet" указывает тип ссылки; в данном случае это ссылка на таблицу стилей;

href="style.css" задает местонахождение внешнего css-файла; значение этого атрибута – url-адрес, который будет формироваться в зависимости от того, где хранится css-файл



ВАЖНО:

К веб-странице можно присоединить некое множество таблиц стилей, добавляя несколько тегов `<link>`, каждый из которых будет указывать на свой файл таблицы стилей



В файле `style.css` прописываем:

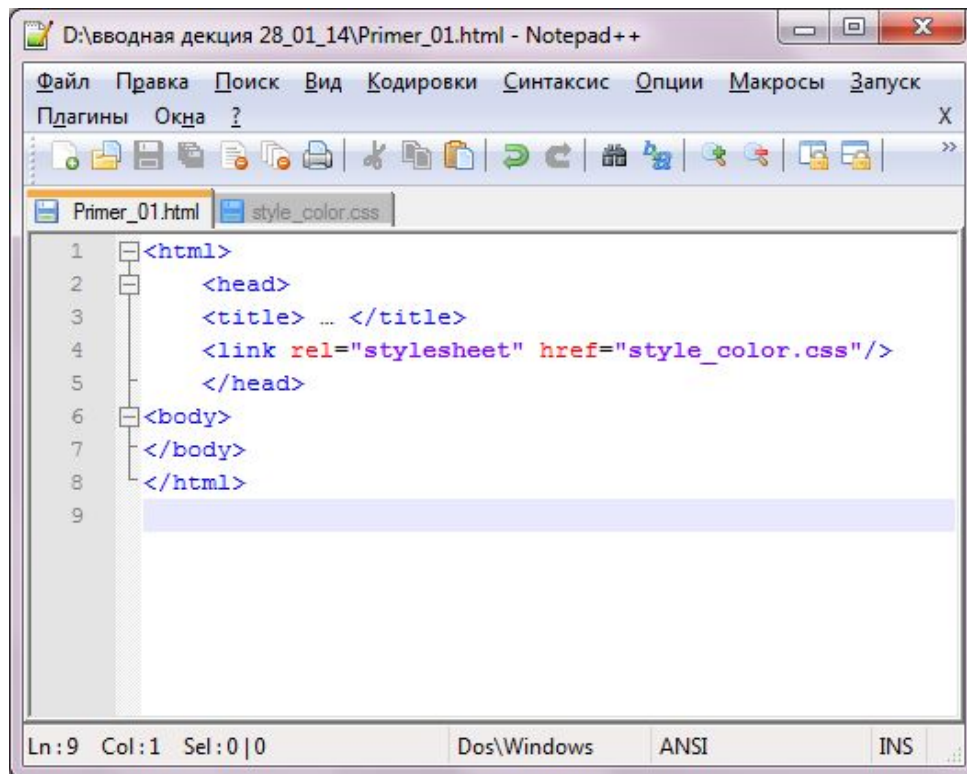
```
html, body
{
background-color: #e87474;    - цвет фона страницы
}
```

```
h1
{
font: 40px Arial;            - для заголовка 1-го уровня
                             размер и тип шрифта
}
```



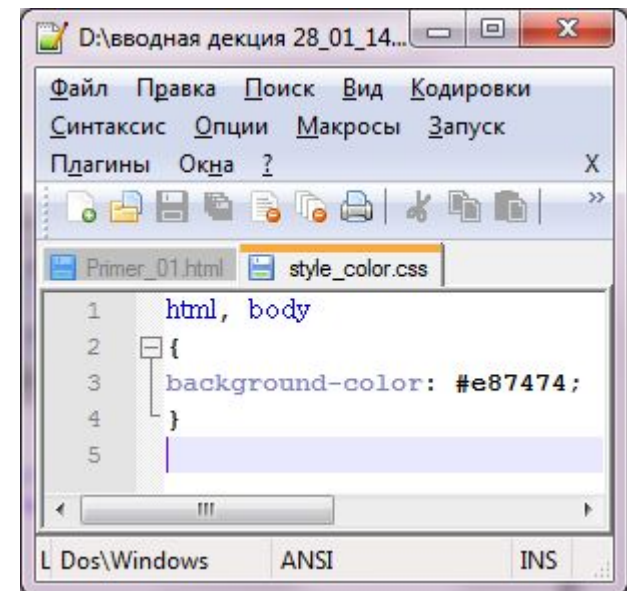
Пример

Файл html-страницы, в которой через ссылку на внешний файл подключены стили для задания цвета фона



```
1 <html>
2   <head>
3     <title> ... </title>
4     <link rel="stylesheet" href="style_color.css"/>
5   </head>
6   <body>
7 </body>
8 </html>
9
```

Файл style_color.css, в котором через стили задан цвет фона страницы



```
1 html, body
2 {
3   background-color: #e87474;
4 }
5
```



Задание

1. Создайте HTML-файл `Primer_color.html`.
2. Через CSS (вынесением в отдельный файл) задайте цвет фона страницы `#a4f4a2`.
3. Сохраните. Просмотрите в браузере результат.
4. Добавьте абзац из нескольких предложений и задайте через стили следующие параметры: цвет текста, размер шрифта.
5. Сохраните. Просмотрите в браузере результат.



Прикрепление таблиц стилей с использованием CSS

CSS имеет встроенный способ привязки внешних таблиц стилей к коду HTML

Для этого в html-код в тег `<style>` нужно добавить следующую запись

```
<style type="text/css">  
    @import url(css/style.css);  
</style>
```



@import – языковая конструкция CSS

url – выполняет привязку через указание пути к стилевому файлу, путь заключать в кавычки необязательно;

через данную языковую конструкцию можно добавить несколько внешних таблиц стилей

после правила @import можно добавлять обычные CSS-стили, если в этом есть необходимость

\ браузеры игнорируют любые таблицы стилей,
импортируемые после CSS-правил



Можно создать внешний css-файл, который содержит только правила **@import**, выполняющие привязку других файлов внешних таблиц стилей; такая методика часто применяется в целях упорядочения и систематизации стилей сайта.



Внутренние таблицы стилей.

Это набор стилей, часть кода веб-страницы, которая всегда должна находиться между открывающим и закрывающим тегами `<style>` HTML-кода в теле тега `<head>`



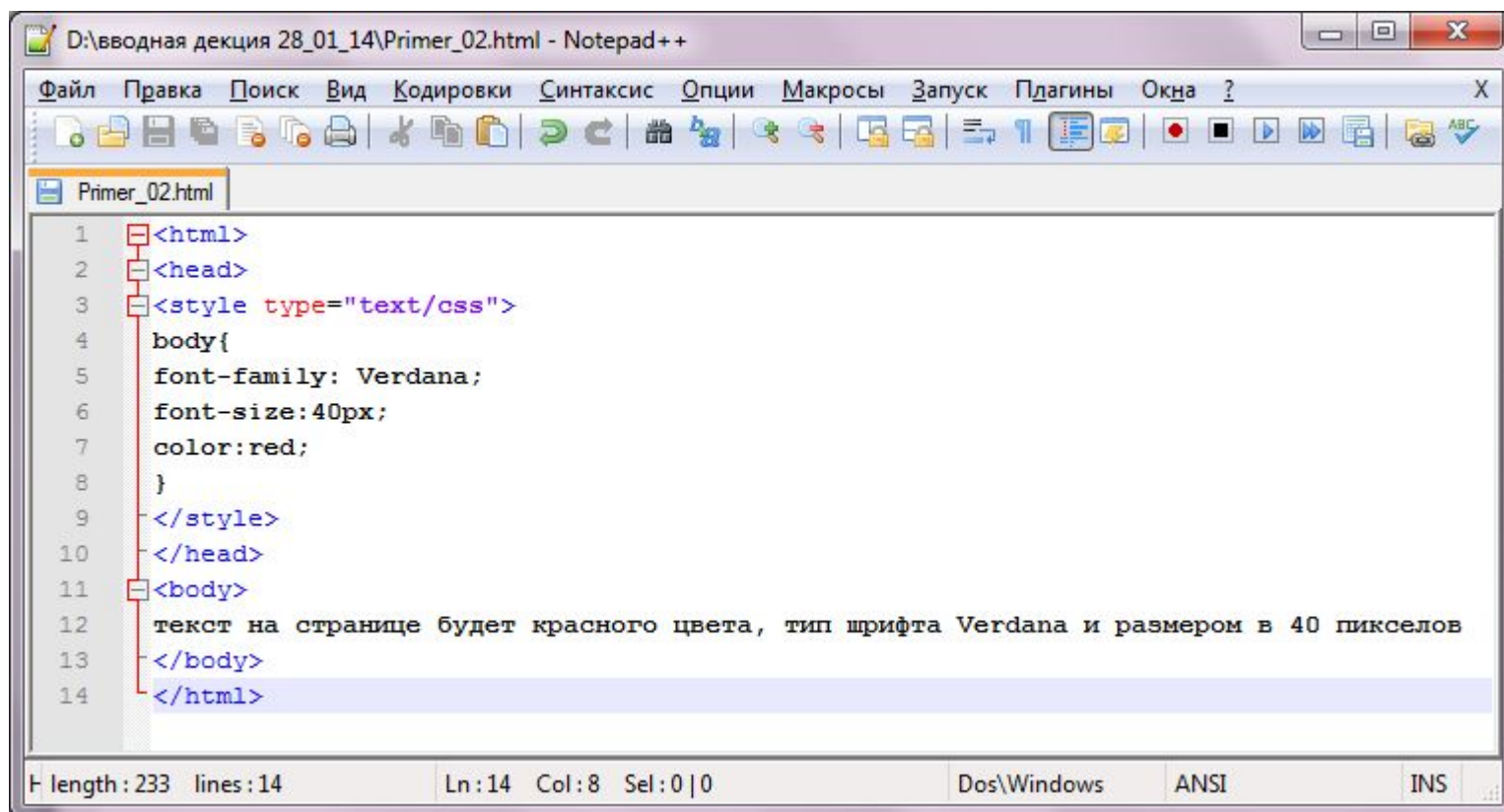
Внутренние таблицы стилей. При включении CSS в исходный файл с помощью тега `<style>`:

```
<html>
<head>
<style type="text/css">
body{
font-family: Verdana;
}
...
</style>
</head>
<body> ...
```



Пример

Стили находятся в самом HTML-файле, в служебной части, заключены в парный тег `<style>...</style>`



The screenshot shows a Notepad++ window titled "D:\вводная декция 28_01_14\Primer_02.html - Notepad++". The menu bar includes "Файл", "Правка", "Поиск", "Вид", "Кодировки", "Синтаксис", "Опции", "Макросы", "Запуск", "Плагины", "Окна", and "?". The toolbar contains various icons for file operations and editing. The main text area shows the following HTML code:

```
1 <html>
2 <head>
3 <style type="text/css">
4   body{
5     font-family: Verdana;
6     font-size:40px;
7     color:red;
8   }
9 </style>
10 </head>
11 <body>
12   текст на странице будет красного цвета, тип шрифта Verdana и размером в 40 пикселей
13 </body>
14 </html>
```

The status bar at the bottom indicates "length: 233 lines: 14", "Ln: 14 Col: 8 Sel: 0 | 0", "Dos\Windows", "ANSI", and "INS".



Задание

1. Создайте HTML-файл Primer_style.html.
2. Через CSS (записать через тег `<style>...</style>`) задайте цвет фона страницы #a2e4f4, цвет шрифта - #04125a, размер шрифта 20 пикселей, тип шрифта – Arial.
3. Сохраните. Просмотрите в браузере результат.



Внутренние таблицы стилей.

Желательно размещать фрагмент кода по стилям прямо перед закрывающим тегом `</head>`.

Если же в код включен JavaScript, то его лучше размещать после таблиц стилей, т.к. зачастую JavaScript полагается на CSS



Внутренние таблицы стилей.

Иногда можно добавить стилевую информацию непосредственно к конкретному HTML-тегу

(примеры по данному способу смотрите в практических работах к лекции)



Спасибо за внимание

