Тема 2.3

«Использование CSS для формирования дизайна веб-страниц»

Вступление

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.

Рассмотрим подробнее…

CSS: общие сведения

CSS используется для задания цветов, шрифтов, расположения и других аспектов представления документа.

Основной целью разработки CSS являлось разделение содержимого (написанного на HTML или другом языке разметки) 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 и печать.

CSS: общие сведения

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

- Стандартный стиль, используемый браузером по умолчанию для представления элементов

CSS: приоритеты

Стандарт CSS определяет приоритеты, в порядке которых применяются правила стилей, если для какого-то элемента подходят несколько правил одновременно.

Это называется <каскадом>, в котором для правил рассчитываются приоритеты или <веса>, что делает результаты предсказуемыми.

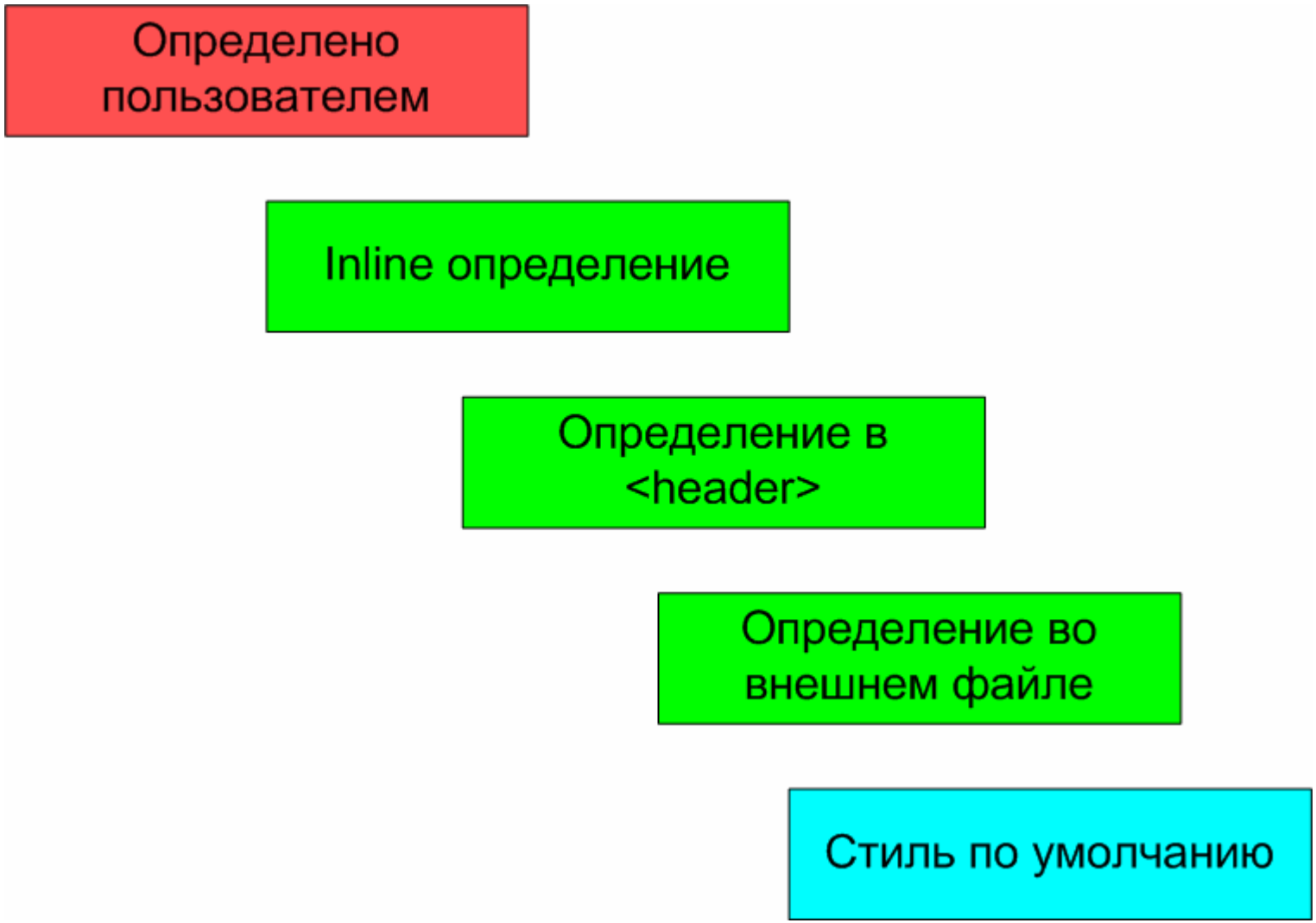
Так, в частности, приоритет inline стиля самый высокий, затем идёт приоритет встроенных стилей, и только затем – приоритет внешний стилей.

Ниже всей этой группы находится стиль браузера по умолчанию.

Выше всей этой группы находится определённый пользователем в браузере стиль.

Для наглядности рассмотрим на рисунке:

CSS: приоритеты



CSS: правила

Таблица стилей состоит из набора правил.

Каждое правило, в свою очередь, состоит из одного или нескольких селекторов, разделённых запятыми, и блока определений.

Блок определений же обрамляется фигурными скобками, и состоит из набора свойств и их значений.

Схематически это можно показать так:

```
селектор1, селектор2
 {
   свойство1: значение;
   свойство2: значение;
   свойство3: значение;
 }
```

Например: `b {color: red}`

```
Просто текст
<b>жирный</b> просто
<br /> текст.
```

Просто текст **жирный** просто
текст.

CSS: группировка селекторов

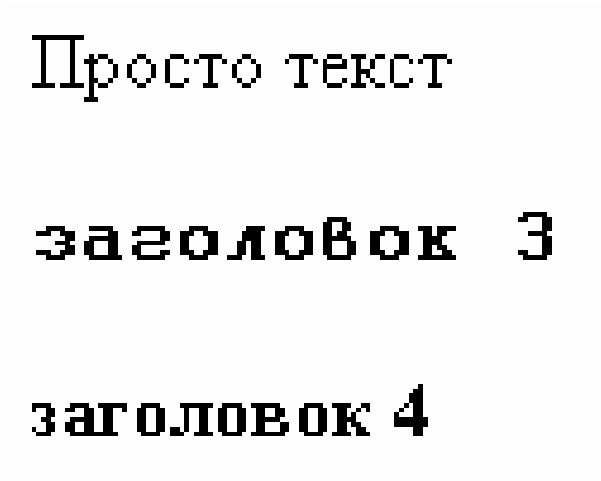
Если один и тот же набор свойств нужно применить к нескольким селекторам, то вместо:

```
h1 {font-family:'Courier'}
h2 {font-family:'Courier'}
h3 {font-family:'Courier'}
```

Можно написать:

```
h1, h2, h3 {font-family:'Courier'}
```

т.е. перечислить селекторы через запятую.



CSS: контекстные селекторы

CSS позволяет определять поведение «элементов, вложенных в другие элементы».

В следующем примере цвет элементов списка будет меняться в зависимости от «глубины вложенности» списка:

```
ol {color: red}
ol ol {color: green}
ol ol ol {color: blue}
ol ol ol ol {color: magenta}
```

1. Уровень 1
    1. Уровень 2
        1. Уровень 3
            1. Уровень 4

А этот пример показывает, что текст, выделенный тегом <b>, находящимся в <I>, и текст, находящийся в теге <u>, вложенном в один тег <p>, будет выделен голубым цветом:

```
b i, p u {color: blue}
```

Просто текст **жирный** ***наклонный***,

Абзац, а там подчёркивание

CSS: специальные селекторы

Рассмотрим следующие примеры:

```
 * {font-weight: bold}
h1 > h3 {font-weight: bold}
h1 + h3 {font-weight: bold}
```

В первом примере звёздочка – универсальный селектор означает, что правило должно быть применено ко всем элементам документа, так что весь текст в нём будет выделен жирным шрифтом.

Во втором примере правило будет применяться ко всем тегам <h3>, являющимся дочерними по отношению к тегу <h1>.

Третий пример иллюстрирует тип смежного селектора, данное правило будет применено к тегам, следующим непосредственно один за другим. В этом случае всякий раз, когда за заголовком первого уровня следует заголовок третьего уровня, заголовок третьего уровня станет жирным.

CSS: псевдоэлементы

Синтаксис псевдоэлемнтов таков:

селектор:псевдоэлемент {свойство:значение;...}

Псевдоэлементы first-line и first-letter определяют стиль первой строки и первой буквы. Например, так вы можете создать «буквицу»:

p:first-letter {font-weight: bold; color: red}

Псевдоэлементы before и after позволяют указывать в документе места, куда вставлять автоматически генерируемое содержимое. Например определив стиль упорядоченного списка следующем образом:

ol:before{content:'Ссылки'}

вы увидите перед каждым списком слово «Ссылки».

Следует учитывать, что псевдоэлементы пока поддерживаются не всеми браузерами и не всегда одинаково.

CSS: регулярные классы

Иногда нужно использовать, например, один стиль абзацев для примеров исходных кодов программ, а другой для описания алгоритмов.

Если ни один из абзацев не будет иметь явного контекста, по которому его можно будет отличить от другого, то можно определить стиль для каждого в отдельности:

```
p.code { font-family: ‘Symbol’; margin-left: 30px; }
p.alg { margin-left:2px; }
```

В селекторе, после имени тега через точку приписано имя класса. Первое правило создало класс стилей абзацев code, текст которых должен выводиться шрифтом Symbol с отступом 30 пикселей от края, а второе – alg, текст которых будет отодвинут от левого края на 2 пикселя. Чтобы применить к содержимому тега определённый класс надо вставить в тег атрибут class и присвоить ему имя стилевого класса.

```
<p class="alg">Текст</p>
```

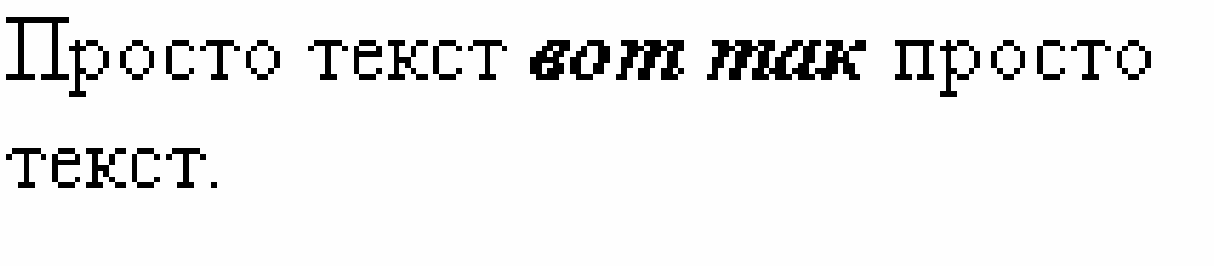
CSS: родовые классы

В рамках стандарта CSS можно создавать классы, не ассоциируя их с каким-нибудь тегом. Например, задав стилевое правило следующем образом:

```
.bold_and_italic {font-style: italic; font-weight: bold}
```

и присвоив bold_and_italic свойству class некоторого тега, вы укажете браузеру, что содержимое этого тега надо отображать жирным и наклонным шрифтом.

```
Просто текст <span class="bold_and_italic">вот так</span> просто текст.
```



CSS: псевдоклассы

Кроме традиционных классов, стандарт CSS определяет ещё и псевдоклассы. Псевдоклассы позволяют управлять отображением элементов, находящихся в каком-нибудь состоянии. Псевдоклассы присоединяются к имени тега двоеточием, и их имена заранее определены:

link, alink, visited – эти псевдоклассы используются только с тегом A и определяют стиль отображения обычной, активной и посещённой ссылок соответственно.

```
a:link{color: blue}
a:active{color: green; font-weight: bold}
a:visited{color: red}
```

hover, focus – это, так называемые, интерактивные классы. Класс hover определяет, как отображать объект, когда на него наведён курсор мыши, а класс focus, определяет, как браузеру показывать содержимое тега, получившему фокус ввода. С помощью этих классов можно украсить ссылки на странице:

```
a:hover{color: yellow}
```

Класс focus не поддерживается большинством браузеров, а класс active работает только с тегом A.

CSS: id-классы

Почти все HTML-теги допускают использование атрибута id. С помощью CSS можно сопоставить тегу с данным значением id некоторое стилевое правило. Имя id-класса отделяется от имени тега знаком #

```
#yellow {color: yellow}
h1#blue {color: blue}
```

Теперь можно создать синий заголовок, написав

```
<h1 id="blue">
```

или, присвоив атрибуту id любого тега значение yellow, окрасить его содержимое в желтый цвет.

Употребление стилей, созданных таким способом обладает существенным недостатком – значение id уникально и документе и, соответственно, правило будет работать только для одного элемента.

CSS: включение стилей в документ

Как уже было сказано, когда мы рассматривали приоритеты стилей, есть три способа включения стилей в HTML-документ:

inline

```
<h1 style="color: blue; font-style: italic">Title</h1>
```

Определение через <head><style> … </style></head>

```
<style type="text/css">
  b {color:red}
</style>
```

Подключение внешних css-документов через <head><link …></head>:

```
<head>
 <link rel="stylesheet" type="text/css" href="http://…/style.css">
</head>
```

CSS: ключевые слова

Значением многих свойств в стандарте CSS могут быть ключевые слова, такие как normal, medium, bold и т.д.

Ключевые слова не чувствительны к регистру, т.е. bold и BOLD – одно и то же.

Каждое свойство может принимать значение inherit, которое указывает, что значение этого свойства надо наследовать у родительского элемента.

CSS: размеры

Если необходимо указать размер того или иного элемента, это можно сделать множеством способов:

em – размер в ширинах буквы m в настоящем шрифте;
ex – размер в высотах буквы x в настоящем шрифте;
px – размер в пикселах;
in – размер в дюймах;
cm – размер в сантиметрах;
mm – размер в миллиметрах;
pt – размер в пунктах (1/72 дюйма);
pc – размер в пиках (12 пунктов).

Например, правило

```
p {text-indent: 3em}
```

задаст красную строку абзаца размером в три буквы m.

Ещё примеры:

```
body {margin: 1in}
table {padding: 1.5cm}
p {text-indent: -2pt}
img.logo {height: 12px}
blockquote {padding-bottom: 2ex}
table {border-width: 7.5pc}
```

CSS: процентные значения

Процентные значения могут быть положительными и отрицательными, а также могут быть десятичными дробями.

Это правило задает высоту строки в абзаце на 20% больше обычной (окружающего текста):

```
p {line-height: 120%}
```

CSS: URL

Значением некоторых свойств может быть URL. Синтаксис в данном случае таков: url("protocol://server/path").

Например:

```
body {background-image: url("/img/bg3.gif")}
```

CSS: цвет

Цвет можно указать с помощью названия (например blue или red – названия определены для «стандартных 16-ти цветов») или с помощью RGB кода.

CSS допускает задание цвета в виде стандартного RGB кода, например: #87FED3.

В отличие от HTML, CSS позволяет задавать цвет трёхзначным числом (т.н. «веб-цвета»), в этом случае каждая цифра числа удваивается при получении реального значения цвета. Таким образом #7C5 тоже самое, что и #77CC55.

Также можно задавать цвет в десятичных rgb обозначениях. Например: rgb (126, 6, 9) или rgb (30%, 40%, 70%).

CSS: краткое описание свойств

Стандарт CSS2 позволяет задавать значение сразу нескольких свойств одним стилевым правилом. То есть конструкцию

```
селектор {
группа_свойств-свойсво1: значение1;
группа_свойств-свойсво2: значение2;
группа_свойств-свойсво3: значение3;}
```

можно заменить на

```
селектор {
группа_свойств: значение1 значение2 значение3}
```

Пример:

```
h1 {font: bold 12px/15px serif}
td {border-left: solid 1px black}
```

Обратите внимание, что значение свойств перечисляются через пробел. Порядок перечисления определяется в каждом конкретном случае отдельно.

CSS: итог

Итак, мы рассмотрели HTML и CSS. Этих знаний вам должно хватить для рассмотрения следующей темы, посвящённой вопросам вёрстки HTML-страниц.