Лекция 6
Каскадные таблицы стилей (CSS). Основы. Синтаксис.

Introduction to Web Development

План лекции

- Что такое CSS?
- Базовый синтаксис
- Применение стилевых правил
- Классы и идентификаторы
- Часто используемые стилевые правила
- Наследование свойств
- Каскадирование
- Специфичность

Что такое CSS?

CSS (Cascading Style Sheets) - каскадные таблицы стилей

Стандарт на основе текстового формата, определяющий представление данных в браузере;

Набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид;

Что такое CSS?

Основной целью разработки CSS являлось **разделение описания логической структуры** веб-страницы (которое производится с помощью HTML или других языков разметки) **от описания внешнего вида этой веб-страницы** (которое теперь производится с помощью формального языка CSS).

Термин "каскадный" означает, что в одной странице HTML могут использоваться разные стили. Браузер, поддерживающий таблицы стилей, будет следовать их порядку (как по каскаду), интерпретируя информацию стилей.

Базовый синтаксис

Селектор — это HTML-элемент, для которого добавляются параметры форматирования. В качестве селектора выступают **теги, классы и идентификаторы.**

Каждое описание содержит **свойство** и **значение**, каждое свойство должно иметь значение.

Стилевые **свойства разделяются** между собой **точкой с запятой**, в конце этот символ можно опустить.

CSS не чувствителен к регистру, переносу строк, пробелам и символам табуляции.

Приоритет имеет значение, указанное в коде ниже.

Базовый синтаксис

Форма записи

Можно записывать так:

```
p {color:red;text-align:center;}
```

```
td { background: olive; }
td { color: white; }
td { border: 1px solid black; }
```

Или так:

```
p {
color:red;
text-align:center;
}
```

```
td {
background: olive;
color: white;
border: 1px solid black;
}
```

Базовый синтаксис

CSS комментарии

Чтобы выделить часть CSS документа как комментарий, используется конструкция /* … */

/* Это комментарий */

```
/*
   Стиль для сайта
*/

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

Применение стилевых правил

Применение стилевых правил

Связанные стили

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Стили</title>
  <link rel="stylesheet" type="text/css" href="mysite.css">
  <link rel="stylesheet" type="text/css" href="main.css">
 </head>
 <body>
   <h1>Заголовок</h1>
   <p>Текст</p>
 </body>
</html>
```

Содержимое файла mysite.css

```
H1 {
  color: #000080;
  font-size: 200%;
  font-family: Arial, Verdana, sans-serif;
  text-align: center;
}
P {
  padding-left: 20px;
}
```

Применение стилевых правил

Глобальные стили

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Глобальные стили</title>
  <style type="text/css">
   H1 {
    font-size: 120%;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    color: #333366;
   }
  </style>
 </head>
 <body>
  <h1>Hello, world!</h1>
 </body>
</html>
```

Применение стилевых правил

Внутренние стили

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Внутренние стили</title>
 </head>
 <body>
  <p style="font-size: 120%; font-family: monospace;
  color: #cd66cc">Пример текста</p>
</body>
</html>
```

Применение стилевых правил

Внутренние стили рекомендуется применять на сайте ограниченно или вообще отказаться от их использования. Дело в том, что добавление таких стилей увеличивает общий объем файлов, что ведет к повышению времени их загрузки в браузере, и усложняет редактирование документов для разработчиков.

Использование **связанных стилей** является наиболее универсальным и удобным методом добавления стиля на сайт. Ведь стили хранятся в одном файле, а в HTML-документах указывается только ссылка на него.

Все описанные методы использования CSS могут применяться как самостоятельно, так и в сочетании друг с другом. В этом случае необходимо помнить об их иерархии**. Первым всегда применяется внутренний стиль, затем глобальный стиль и в последнюю очередь связанный стиль.**

Классы и идентификаторы

Классы применяют, когда необходимо определить **стиль для индивидуального элемента веб-страницы** или **задать разные стили для одного тега**.

Тег.Имя класса { свойство1: значение; свойство2: значение; ... }

```
<style type="text/css">
  P { /* Обычный абзац */
   text-align: justify; /* Выравнивание текста по ширине */
  }
  P.cite { /* Абзац с классом cite */
   color: navy; /* Цвет текста */
   margin-left: 20px; /* Отступ слева */
   border-left: 1px solid navy; /* Граница сле
   padding-left: 15px; /* Расстояние от линии
  }
 </style>
```

```
<body>
  <p>Обычный абзац</p>
  <p class="cite">Абзац со стилем</p>
 </body>
```

Классы и идентификаторы

Можно, также, использовать **классы без указания тега:**

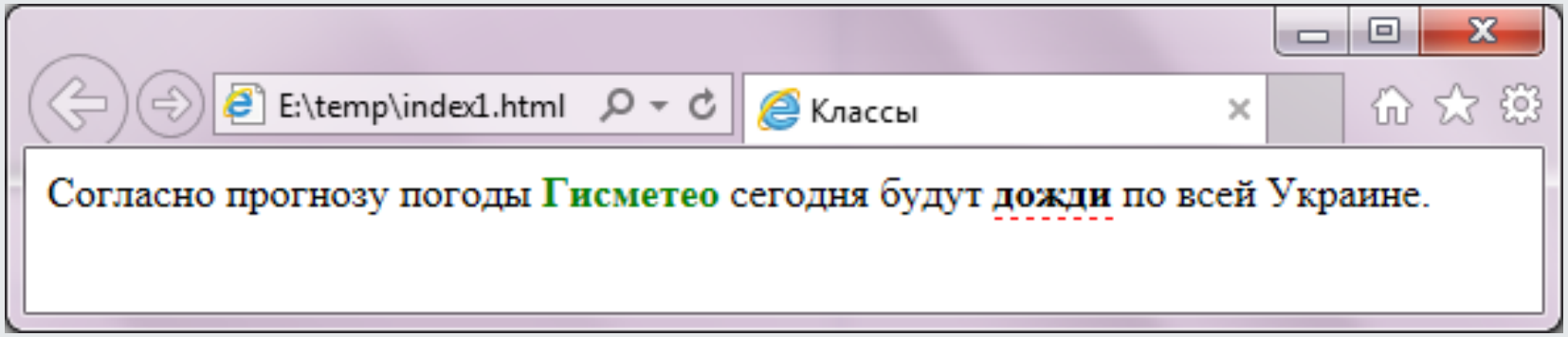
.Имя класса { свойство1: значение; свойство2: значение; ... }

При такой записи, класс можно применять к любому тегу:

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Классы</title>
  <style type="text/css">
   .gost {
    color: green; /* Цвет текста */
    font-weight: bold; /* Жирное начертание */
   }
   .term {
    border-bottom: 1px dashed red; /* Подчеркивание под текстом */
   }
  </style>
</head>
<body>
   <p>Согласно прогнозу погоды <span class="gost">Гисметео</span> сегодня будут
      <b class="term">дожди</b> по всей Украине.
   </p>
</body>
```

Классы и идентификаторы

Результат в браузере:



Классы **удобно использовать**, когда нужно **применить стиль** к разным элементам веб-страницы: **ячейкам таблицы, ссылкам, абзацам и др.**

К любому тегу **одновременно можно добавить несколько классов**, перечисляя их **в параметре class через пробел**. В этом случае к элементу применяется стиль, описанный в правилах для каждого класса.

Классы и идентификаторы

Идентификатор («ID селектор»)

определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты.

#Имя идентификатора { свойство1: значение; свойство2: значение; ... }

Обращение к идентификатору происходит аналогично классам, но в качестве ключевого слова у тега используется параметр id, значением которого выступает имя идентификатора.

<тег id="Имя идентификатора">

Идентификаторы **можно применять к конкретному тегу**:

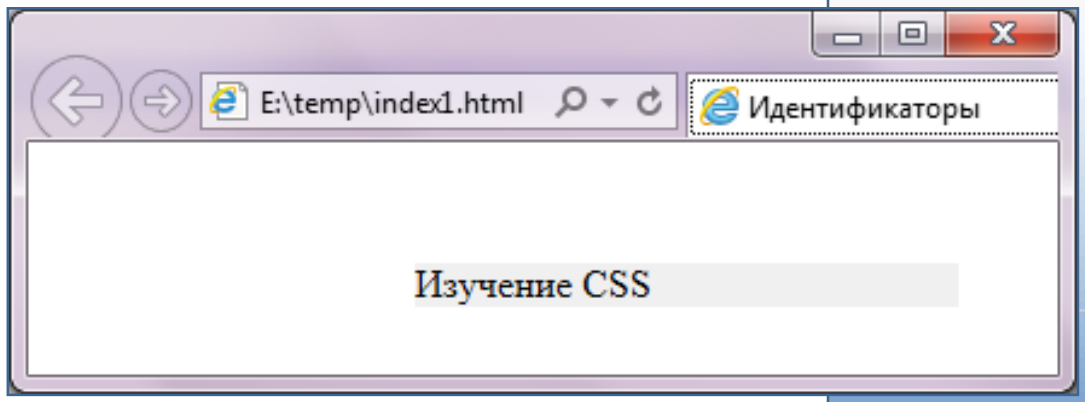
Тег#Имя идентификатора { свойство1: значение; свойство2: значение; ... }

Классы и идентификаторы

Пример использования идентификатора:

```
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
 <title>Идентификаторы</title>
 <style type="text/css">
  #help {
   position: absolute; /* Абсолютное позиционирование */
   left: 160px; /* Положение элемента от левого края */
   top: 50px; /* Положение от верхнего края */
   width: 225px; /* Ширина блока */
   padding: 5px; /* Поля вокруг текста */
   background: #f0f0f0; /* Цвет фона */
  }
 </style>
</head>
<body>
 <div id="help">
  Изучение CSS!
 </div>

</body>
```



Классы и идентификаторы

Идентификаторы

- В коде документа каждый идентификатор **уникален** и **должен быть включён лишь один раз.**
- Имя идентификатора **чувствительно к регистру**.
- Через метод **getElementById** можно получить доступ к элементу по его идентификатору и изменить свойства элемента.
- Стиль для идентификатора имеет **приоритет выше, чем у классов.**

Классы

- Классы могут использоваться в коде **неоднократно**.
- Имена классов **чувствительны к регистру**.
- Классы можно комбинировать между собой, добавляя **несколько классов к одному тегу**.

Основные CSS свойства

Свойство	Название
Цвет	color, background-color
Интервал	margin, padding, margin-left, margin-right, margin-top, margin-bottom
Границы	border-width, border-style, border-color, border (для установки ширины, стиля и цвета границ за один прием)
Выравнивание текста	text-align, text-indent, word-spacing, letter-spacing, line-height, white-space
Шрифты	font-family, font-size, font-weight, font-style, font-variant, text-decoration, @font-face (для использования вычурных шрифтов)
Размер	width, height
Позиция	position, left, right, float, center
Графика	background-image, background-repeat, background-position

Основные CSS свойства

Для форматирования текста

color

Данный стиль **задает цвет текста**. Для задания цвета можно использовать как хекс-значение цвета (*color:#FFF*), так и ряд ключевых слов (*color:black, color:red …*)

text-align

Данный стиль **задает выравнивание текста внутри родительского блока**. Может иметь значения *left, right, center*. Есть еще значение стиля - *justify,* которое выравнивает текст по всей ширине родительского блока. Среди веб-дизайнеров стиль *justify* считается плохим тоном, т.к. выравнивание по всей ширине родительского блока приводит к появлению пробелов различной длины, что сильно ухудшает читабильность.
Выравнивание, заданное свойством *text-align,* распространяется так же на графические элементы внутри блока.

Основные CSS свойства

Для форматирования текста

line-height

Данный стиль задает **расстояние между строк в текстовом блоке** (или, иными словами, изменяет высоту строки текста, еще это называется «интерлиньяж»). Порой шрифт значительно приятнее смотрится при увеличении значения *line-height*, заданного по умолчанию.
Значение данного свойства задается в процентах (100%, 150% …), множителем (1 - интерлиньяж по умолчанию, 1.5 — увеличен в полтора раза) или точным значением в пикселах (10px, 1.5 em…).

letter-spacing

Межсимвольное расстояние. Значение данного свойства указывает в единицах длины (пиксели, дюймы, pt), либо относительные единицы — em.

Основные CSS свойства

Для форматирования текста

font

Универсальное свойство, которое позволяет одновременно задать несколько характеристик шрифта и текста.

В качестве **обязательных** значений свойства font указывается **размер шрифта и его семейство**. Остальные значения являются опциональными и задаются при желании. Для подробного ознакомления смотрите информацию о каждом свойстве отдельно.

font: [font-style||font-variant||font-weight]
font-size [/line-height] font-family | inherit

```
p { font: 12pt/10pt sans-serif; }
```

```
p { font: normal small-caps 12px/14px fantasy;
```

```
p { font: bold italic 110% serif; }
```

Основные CSS свойства

Для форматирования текста

font-family

Данный стиль служит для **задания гарнитуры шрифта**. Название шрифтов перечисляются через запятую, в случае, если название состоит более чем из одно слово необходимо использовать кавычки.
font-family: (шрифт без засечек), Arial, Helvetica, sans-serif;
font-family: (шрифт с засечками), «Times New Roman», Times, serif;
font-family: (моноширинный шрифт), «Courier New», Courier, monospaced;

font-weight

При желании сделать текст блока **жирным** — используйте стиль ***font-weight:bold.***
Если вы наоборот хотите **убрать жирное выделение** — тут все просто ***font-weight:normal***

Основные CSS свойства

Для фона

В качестве фона можно задать картику: ***background-image***;
либо просто задать фоновый цвет: ***background-color***.
Фон может повторяться (по оси X или Y) - ***background-repeat***.
А так же фон можно смещать — ***background-position***.

CSS2.1	background: [background-attachment || background-color || background-image || background-position || background-repeat] | inherit
CSS3	background: [<фон>,]* <последний_фон>

```
background: url(images/hand.png) repeat-y #fc0;
```

Основные CSS свойства

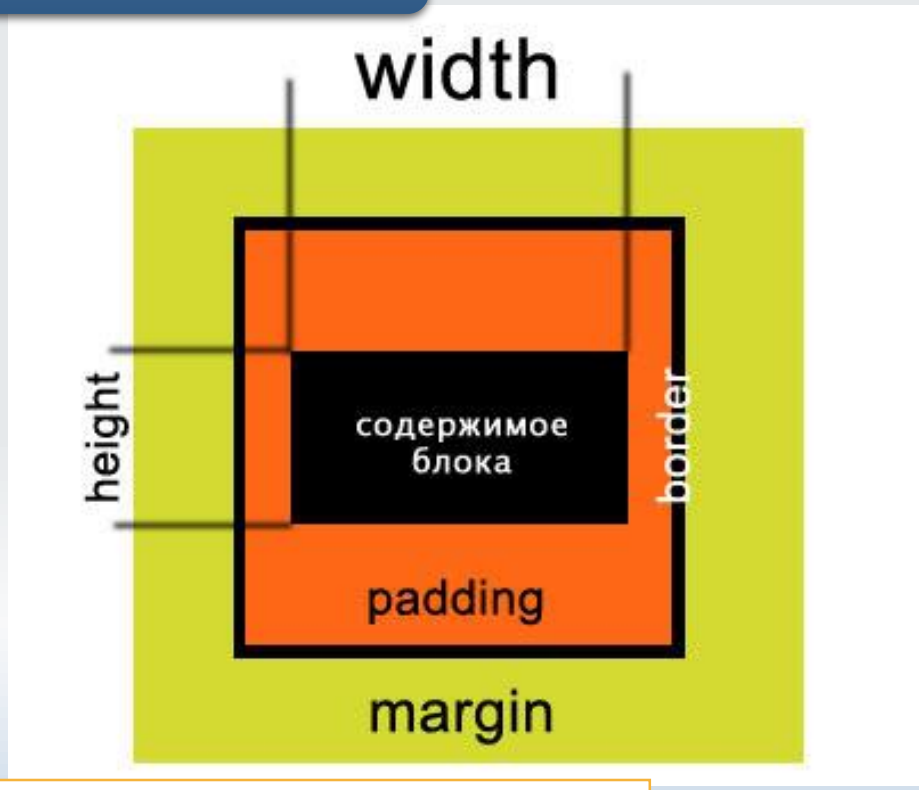
Для отступов

padding

Отступ внутри блока (значение в пикселах)

margin

Отступы от границ блока



***{padding:0; margin:0}**

Данный стиль убирает отступы, задаваемые в браузере по умолчанию.
Рекомендовано начинать написание файла CSS именно с этого стиля.

Основные CSS свойства

Позиционирование элементов

```
position: absolute | fixed | relative | static | inherit
```

position:absolute

Данное свойство вырывает блок из нормального потока формирования страницы. И далее блок позиционируется относительно верхнего угла браузера путем указания свойств *left* и *top* в процентах или пикселях.

Нормальный поток — это когда блоки идут на странице один за одним, т.е. div под div-ом. В случае применения свойства *position:absolute* блок накладывается поверх остальных блоков нормального потока.

Используя абсолютное позиционирование, дабы избежать непредвиденных обстоятельств, необходимо задать значения ширины и высоты блока в пикселях (свойства *width* и *height)*.

Основные CSS свойства

Позиционирование элементов

position:relative

Разновидность абсолютного позиционирования. В данном случае блок смещается заданием значений *left* и *top* относительно места своего нормального положения.

Иными словами, блок выводится там, где он и должен быть в нормальном потоке и сдвигается на заданные значения.

Если у родительского блока указан стиль *position:relative*, то вложенный блок с указанным стилем *position:absolute* будет смещаться относительно левого верхнего угла родительского блока.

Основные CSS свойства

Позиционирование элементов

float:left

В случае, если необходимо разместить два блока *div* в одну линию друг за другом, у первого блока указывается стиль *float:left* (это означает что своей левой границей данный блок должен прилипнуть к предыдущему блоку в потоке). Первый блок оказывается прижатым, например к левой границе окна браузера. Если следующему блоку в потоке указать тоже самое значение в стилях, то два блока будут выводиться на одной линии. Первый блок будет прилипать к левой границе экрана, а второй, своей левой границей, к правой границе предыдущего блока.

Если у второго блока указать значения стиля *float:right*, то оба блока все так же окажутся расположенными на одной линии, но теперь первый блок будет прилипать к левой границе окна браузера, а правый — своей правой стороной к правой границе окна браузера.

Наследование свойств

Наследование - перенос правил форматирования для элементов, находящихся внутри других.
Такие элементы являются дочерними, и они наследуют некоторые стилевые свойства своих родителей, внутри которых располагаются.

Наследование **позволяет задавать значения некоторых свойств единожды, определяя их для родителей верхнего уровня**.

Допустим, требуется установить цвет и шрифт для основного текста:

```
 <style type="text/css">
  BODY {
   font-family: Arial, Helvetica, sans-serif; /* Гарнитура шрифта */
   color: navy; /* Синий цвет текста */
  }
 </style>
...
<body>
 <p>Цвет текста этого абзаца синий.</p>
</body>
```

Каскадирование

Каскадирование - одновременное применение разных стилевых правил к элементам документа — с помощью подключения нескольких стилевых файлов, наследования свойств и других методов.

Приоритеты (в списке по нарастанию):

- **Стиль браузера**
- **Стиль автора**
- **Стиль пользователя**
- **Стиль автора с добавлением !important**
- **Стиль пользователя с добавлением !important**

Специфичность

Если к одному элементу **одновременно применяются противоречивые стилевые правила**, то **более высокий приоритет** имеет правило, у которого **значение специфичности селектора больше**.

Специфичность - это условная величина, вычисляемая следующим образом:

- за каждый **идентификатор** начисляется **100** (в примере - a)
- за каждый класс и **псевдокласс** начисляется **10** (в примере - b)
- за каждый **селектор тега** и **псевдоэлемент** начисляется **1** (в примере – c).

Складывая указанные значения в определенном порядке, получим значение специфичности для данного селектора.

```
ul ol+li          {} /* a=0 b=0 c=3 -> специфичность = 3 */
ul li.red         {} /* a=0 b=1 c=2 -> специфичность = 12 */
li.red.level      {} /* a=0 b=2 c=1 -> специфичность = 21 */
#t34              {} /* a=1 b=0 c=0 -> специфичность = 100 */
```

Встроенный стиль, добавляемый к тегу через атрибут style, имеет специфичность 1000, поэтому всегда перекрывает связанные и глобальные стили. Однако **добавление !important перекрывает в том числе и встроенные стили**.

Рекомендованная литература:

- https://www.w3.org/
- http:// www.w3schools.com/
- http://htmlbook.ru/
- https://www.codecademy.com
- https://htmlacademy.ru/
- Новая большая книга CSS

и…

http://www.google.com/