ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ КОМПЬЮТЕРНЫЕ ЯЗЫКИ РАЗМЕТКИ

ЖИЛЯК
НАДЕЖДА АЕКСАНДРОВНА

Стоимость проведения лабораторной работы

190800

CSS

Cascading Style Sheets

Стилем или CSS

(Cascading Style Sheets, каскадные таблицы стилей) ***называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид***

CSS представляет собой мощную систему, расширяющую возможности дизайна и верстки веб-страниц

CSS

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

Включение в HTML

Внешний файл:

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

Непосредственно в HTML-документе:

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

Непосредственно в элемент:

```
<p style="font-size: 21px; color: green;">Рассказ о том, как вредно красить батареи</p>
```

CSS

Табл
очере
запят
фигу

Схем

селек
  сво
  сво
  сво
}

```
p {
  font-family: "Garamond", serif;
}
h2 {
  font-size: 110 %;
  color: red;
  background: white;
}
.note {
  color: red;
  background: yellow;
  font-weight: bold;
}
p#paragraph1 {
  margin: 0;
}
a:hover {
  text-decoration: none;
}
#news p {
  color: blue;
}
```

Типы верстки

На сегодняшний день существует два основных типа верстки веб-документов:

- табличная верстка (посредством HTML-таблиц)
- блочная верстка (посредством CSS)

CSS-верстка

Преимущества:

- Несколько дизайнов страницы для разных устройств просмотра.
- Уменьшение времени загрузки страниц сайта
- Простота последующего изменения дизайна.
- Дополнительные возможности оформления.
- Оптимально с точки зрения поисковых систем.

Недостатки:

- Различное отображение вёрстки в различных браузерах (особенно устаревших)
- Часто встречающаяся необходимость на практике исправлять не только один CSS-файл, но и теги HTML, которые сложным и ненаглядным способом связаны с селекторами CSS

Уровень 1 (CSS1)

Рекомендация W3C принята 17 декабря 1996, откорректирована 11 января 1999. Среди возможностей, предоставляемых этой рекомендацией были:

Параметры шрифтов. Возможности по заданию гарнитуры и размера шрифта, а также его стиля - обычного, курсивного или полужирного.

Цвета. Спецификация позволяет определять цвета текста, фона, рамок и других элементов страницы.

Атрибуты текста. Возможность задавать межсимвольный интервал, расстояние между словами и высоту строки (т.е. межстрочные отступы)

Выравнивание для текста, изображений, таблиц и других элементов.

Свойства блоков, такие как высоту, ширину, внутренние (padding) и внешние (margin) отступы и рамки. Так же в спецификацию входили ограниченные средства по позиционированию элементов, такие как float и clear.

И другое...

Уровень 2 (CSS2)

Рекомендация W3C принята 12 мая 1998. Построена на CSS1 с сохранением обратной совместимости. Добавление функциональности:

Блочная верстка. Появились относительное, абсолютное и фиксированное позиционирование. Позволяет управлять размещением элементов по странице без табличной верстки

Типы носителей. Позволяет устанавливать разные стили для разных носителей (например монитор, принтер, КПК)

Звуковые таблицы стилей. Определяет голос, громкость и т. д. для звуковых носителей (например для слепых посетителей сайта)

Страничные носители. Позволяет, например, установить разные стили для элементов на чётных и нечётных страницах при печати

Расширенный механизм селекторов

Генерируемое содержание. Позволяет установить текст или картинку, который будет отображаться до или после нужного элемента

И другое…

Уровень 2.1 (CSS2.1)

Рабочая версия W3C от 6 ноября 2006. Построена на CSS2, содержит исправления ошибок

Уровень 3 (CSS3)

Сильно расширена по сравнению с предыдущими версиями. Нововведения, начиная с малых, вроде закругленных углов блоков, заканчивая трансформацией (анимацией) и, возможно, введением переменных.

CSS: влияние на технологию

1. Сначала нужно определиться с номенклатурой страниц, т.е. все страницы проектируемого Web-узла разбить на типы, например,: домашняя страница, навигационные страницы, информационные страницы, коммуникационные страницы и т. п.. У каждого узла этот перечень может быть своим.
2. Для каждого из типов страниц разрабатывается определенная логическая структура (стандартный набор компонентов страницы).
3. После этого разрабатывается навигационная карта узла и форма ее реализации на

CSS: влияние на технологию

4. Для каждого стандартного компонента страницы разрабатывается стиль его отображения (CSS-описатель).
5. Теперь остается только рисовать картинки, создавать анимацию, писать программы, вручную вводить текст и графику или генерировать содержание страниц автоматически во время обращения к ним.

Способы применения CSS

- переопределение стиля в элементе разметки
- размещение описания стиля в заголовке документа в элементе STYLE
- размещение ссылки на внешнее описание через элемент LINK
- импорт описания стиля в документ

L/O/G/O

Переопределение стиля

Под переопределением стиля в элементе разметки мы понимаем применение атрибута STYLE у данного элемента разметки:

```
<h1 style="font-weight:normal;
font-style:italic;
font-size:10pt;">
Заголовок первого уровня
</h1>
```

L/O/G/O

Элемент STYLE

Элемент STYLE позволяет определить стиль отображения для:

- стандартных элементов HTML-разметки
- произвольных классов (селектор class)
- HTML-объектов (селектор id)

L/O/G/O

Пример

```
<head>
<style>
p {color:darkred;text-align:justify;
  font-size:8pt;}
</style>
</head>
<body>
...
<p>Этот параграф мы используем в качестве
  примера применения описания стиля для
  стандартного элемента HTML-
  разметки.</p>
</body>
```

Ссылка на описание стиля

Ссылка на описание стиля, расположенное за пределами документа, осуществляется при помощи элемента LINK, который размещают в элементе HEAD.

```
<link type="text/css"
  rel="stylesheet"
  href="http://kuku.ru/my_css.css
  " >
```

Импорт стиля

Импортировать стиль можно либо внутрь элемента STYLE, либо внутрь внешнего файла, который представляет собой описатель стиля. Оператор импорта стиля должен предшествовать всем прочим описателям стилей:

```
<style>
@import:url(http://kuku.ru/style
.css)
a
{color:cyan;text-decoration:unde
rline;}
```

Синтаксис

Синтаксис описания стилей в общем виде представляется следующим образом:

```
selector[, selector[, ...]]{attribute:v
alue;
 [atribute:value;...]}
```

или

```
selector selector [selector ...]
 {attribute:value;[atribute:value;...]}
```

Первый вариант перечисляет селекторы, для которых действует данное описание стиля. Второй вариант задает иерархию вложенности селекторов, для совокупности которых определен стиль.

Описание селектора

Селектор - имя элемента разметки

```
i, em {color:#003366,font-style:normal}
a, i {font-style:normal;font-weight:bold;
   text-decoration:line-through}
```

Первая строка этого описания перечисляет селекторы-элементы, которые будут отображаться одинаково:

```
<i>Это курсив</i> и это тоже
   <em>курсив</em>
```

Это курсив и это тоже *курсив*

Последняя строка определяет стиль отображения вложенного в гипертекстовую ссылку курсива:

```
<a name=empty><i>kuku</i></a>
```

kuku

Селектор - имя класса

```
<style>
.kuku
  {color:darkred;background-color:white;}
</style>
<p class=kuku>Этот параграф мы отобразим
  темно-красным цветом по белому фону.</p>
<p>Эту <a class=kuku>гипертекстовую
  ссылку</a> мы отобразим темно-красным
  цветом по белому фону.</p>
```

Этот параграф мы отобразим темно-красным цветом по белому фону.

Эту гипертекстовую ссылку мы отобразим темно-красным цветом по белому фону.

Селектор - идентификатор объекта

Описание стиля для объекта задается строкой, в которой селектор представляет собой имя этого объекта с лидирующим символом "#":

```
a.mainlink
{color:darkred;text-decoration:underline;font-style:italic;}
#blue {color:#003366}
<a class=mainlink>основная гипертекстовая ссылка</a>
<a class=mainlink id=blue>модифицированная гипертекстовая ссылка</a>
```

основная гипертекстовая ссылка

модифицированная гипертекстовая ссылка

Наследование и переопределение

- Сначала применяются стили умолчания браузера
- Стили умолчания браузера переопределяются прилинкованными стилями (элемент LINK заголовка документа).
- Прилинкованные стили переопределяются описаниями стилей в элементе STYLE
- Стили элемента STYLE переопределяются атрибутом STYLE в любом из элементов разметки

Элемент DIV

DIV позволяет применить атрибуты стиля, связанные с границей блокаDIV позволяет применить атрибуты стиля, связанные с границей блока, отступами блока от границ старшего элементаDIV позволяет применить атрибуты стиля, связанные с границей блока, отступами блока от границ старшего элемента и "набивку", т.е. отступ от границы блока до границы вложенного элемента:

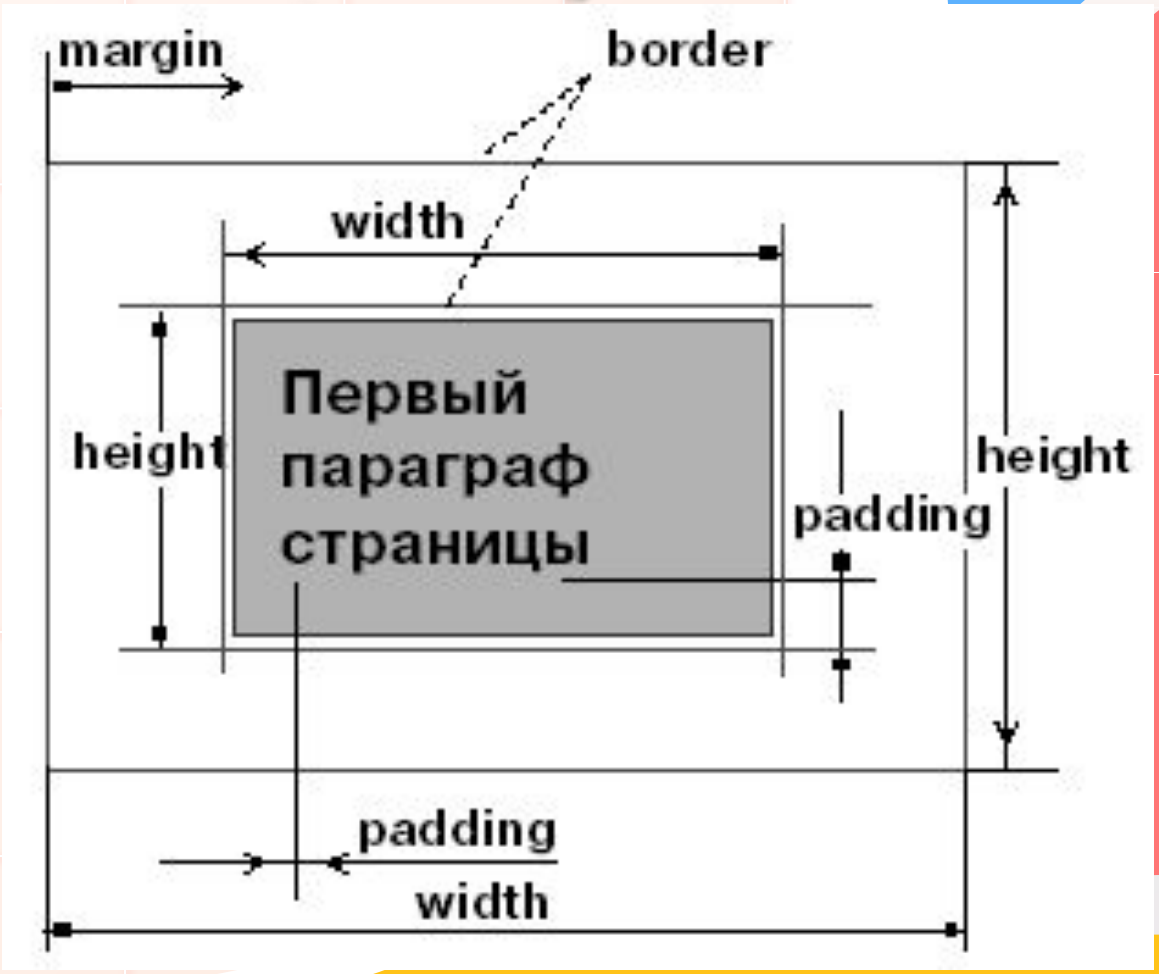
```
<div style="border-
color:#003366;
 border-width:1px;
 margin:20px;
```

Элемент SPAN

Элемент разметки SPAN - это обобщенный строковый элемент разметки, применение которого не приводит к образованию блока текста. Он может заменить собой элементы: FONT, I, B, U, SUB, SUP и т.п.

```
<span style="font-
weight:bold;">Стили
</span style="font-
style:italic;">могут </span>
пересекаться</span>
```

Свойства блоков



L/O/G/O

Отступы (margins)

```
p
  {margin-left:50px;margin-right:5px
  ;
  margin-top:15px;margin-bottom:50px
  ;
                                  gn:left;}
```

Набивка (padding)


```
p
  {padding-left:10
  0px;
  padding-right:50
  px;
  padding-top:20px
  ;
  padding-bottom:1
  0px;
  text-align:left;
  border-width:1px
  ;}
```

Обтекание блока текста

Атрибут float определяет плавающий блок текста. Он может принимать значения:

- left - блок прижат к левой границе охватывающего блок элемента;
- rigth - блок прижат к правой границе охватывающего блок элемента;
- both - текст может обтекать блок с обеих сторон.

Атрибут clean – запрет обтекания:

```
<p
  style='clear:right;text-align:justify;
  '>У этого блока запрещен \"плавающий\"
  блок справа, поэтому он начинается
  ниже прижатого вправо ограниченного
  блока.</p>
```

Управление цветом

Цвет текста:

```
td {color:darkred;}
p {color:darkred;}
i {color:#003366;font-style:normal;}
```

Цвет фона:

```
<body bgcolor=...>...</body>
<table bgcolor=...>...</table>
<span style="background-color:#003366;
  color:white;">как строковые элементы
  разметки</span>
p {background: gray
  http://kuku.ru/kuku.gif no-repeat
  fixed center center;}
```

Шрифты

- font-family - семейство начертаний шрифта (гарнитура);
- font-style - прямое начертание или курсив;
- font-weight - "усиление" (насыщенность) шрифта, "жирность" букв;
- font-size - размер шрифта (кегль). Задается в пикселях (px) и типографских пунктах (pt);
- font-variant - вариант начертания (обычный или мелкими буквами - капитель).

Гарнитура

```
<p align=left style="font-size:24px;
  font-family:serif;color:darkred;">Эта
  строка набрана пропорциональным шрифтом
  с засечками.</p>
<p align=left style="font-size:24px;
  font-family:sans-serif;color:darkred;">Э
  та строка набрана пропорциональным
  шрифтом без засечек.</p>
<p align=left style="font-size:24px;
  font-family:monospace;color:darkred;">Эт
  а строка набрана моноширинным
  шрифтом.</p>
```

Кегль (font-size)

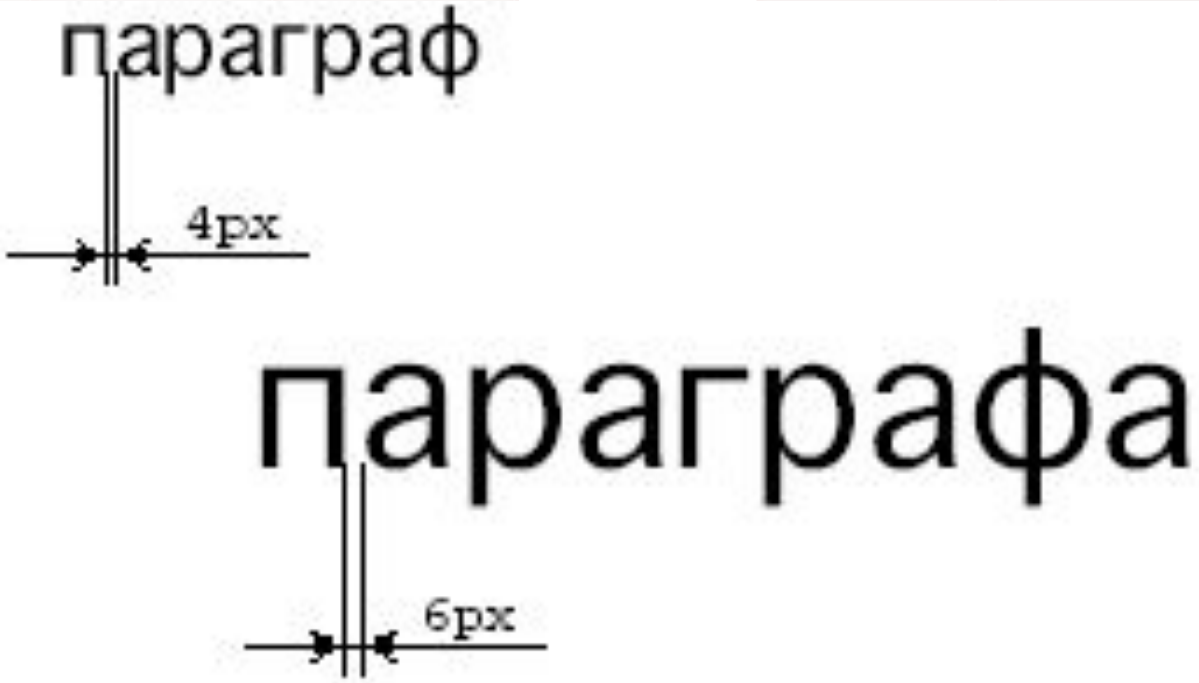
```
<p style="font-size:12pt;">Кегль
  параграфа установлен в 12 пунктов</p>
<p style="font-size:12px;">Кегль
  параграфа установлен в 12 пикселей</p>
<p style="font-size:120%;">Кегль
  параграфа установлен в 120% от размера
  букв охватывающего параграф
  элемента</p>
<p style="font-size:large;">Размер кегля
  large</p>
```

Начертание

```
<p style="color:darkred;
  font-style:normal;">Прямое
  начертание</p>
<p style="color:darkred;
  font-style:italic;">Курсив</p>
<p
  style="color:darkred;font-style:itali
  c;
  font-weight:bold;">Курсив</p>
```

Межбуквенные расстояния

```
<p style="font-family:monospace;
letter-spacing:10pt;color:darkred">
Межбуквенное расстояние 10pt</p>
```



Визуальные эффекты в каскадных таблицах стилей

- Каждый блок-бокс ограничен в размерах, например шириной и высотой окна браузера или если вы явно задали размер какого либо бокса, при этом количество контента превышает эти размеры то происходит **переполнение**, т.е. часть содержимого прячется и чтобы нам увидеть спрятанный контент достаточно прокрутить полосу прокрутки.

- В каскадных таблицах стилей мы можем управлять визуальными эффектами, в том числе управлять полосой прокрутки, сжатием или даже обрезкой страницы до размера бокса.

Для начала рассмотрим ситуации когда появляется полоса прокрутки:

- 1)**Переполнение по высоте**, эта ситуация возникает если контента на странице больше чем размер окна браузера.
  2)**Переполнение по ширине** возникает если контент не может быть перенесен на другую строку, например текст в теге PRE или слишком широкая картинка или другой объект
  3)Явно заданные размеры бокса, но контент в него не помещается.
  4)Если **бокс позиционирован** абсолютно (class{position:absolute})
  5) Если задан **отрицательный размер бокса**.

Чтобы управлять переполнением, т.е. полосой прокрутки в каскадных таблицах стилей существует существует правило overflow, которое может принимать следующие значения:

- 1)**class{overflow:visible}** - это значение по умолчанию, его задавать не обязательно кроме тех случаев если вам нужно вернуть умолчания. При использовании этого правила полоса прокрутки будет возникать только при необходимости, т.е. если действительно будет происходить переполнение блока.
  2)**class{overflow:hidden}** - это правило скрывает полосу прокрутки всегда, даже если происходит переполнение и скрытое содержимое не отображается ни когда.
  3)**class{overflow:scroll}** в этом случае полоса прокрутки будет всегда, независимо происходит переполнение или нет. Очень полезное правило при динамическом содержимом.
  4)**class{overflow:auto}** аналог overflow:visible, но зависит от браузера.
  5)**class{overflow:inherit}** на усмотрение браузера(пользовательского агента).

•

Свойство overflow не наследуется.

- По умолчанию браузер выводит на экран контент с верхнего левого угла и сколько влезет,
- в каскадных таблицах стилей мы можем изменять умолчания при помощи свойства **clip** и отображать нужную нам область.

Свойство правила clip принимает следующие значения:

- 1)**class{clip:auto}** Сжимаемая область имеет тот же размер и размещение, что и боксы элемента.
  2)**class{clip:inherit}** На усмотрение браузера.
  3)**class{clip:rect(верх справа низ слева)}**
- В этом значении мы сами задаем область видимости, значение могут быть **auto** или единицы длины, например **class{clip:rect(auto 10px -20px 50px)}**
  Свойство **clip** применяется дополнением к свойству **overflow**, только в том случае если значение последнего отлизается от overflow:visible.
  Свойство **clip** не наследуется.

- Свойство правил **visibility** предписывает отображать ли бокс или нет.
- Например если вернуться к теме об моделях боксов и вспомнить **display: none**, не смотря на то что блок с таким правилом скрыт, он все же занимает место на странице.
- А свойство **visibility** управляет этим процессом, т.е. резервирует место под боксы или вообще подавляет.

Свойство visibility принимае т такие значения:

- 1)**class{visibility:visible}**
- Место под данный бокс резервируется. 2)**class{visibility:hidden}**
- Место под данный бокс вырезается. 3)**class{visibility:collapse}**
- Если это значение не используется для тегов TR, TH и TD то используется как и visibility:hidden.

Звуковые Таблицы стилей

не все пользователь могут четко видеть ваш дизайн, а некоторые вообще его не видят, а причина вся в том что у них ***очень плохое зрение*** и для таких пользователей существуют **звуковые браузеры**, которые считывают текст с страниц и воспроизводят их

В CSS есть целый ряд правил, который поможет правильно обработать ваши файлы

- Правило **volume:** позволяет регулировать громкость и имеет значения:
  volume:50 - это значение устанавливается цифрой в диапазоне от нуля до ста, соответственно чем цифра больше тем громче.
  volume:silent - без звука.
  volume:x-soft - аналогично volume:0.
  volume:soft - аналогично volume:25.
  volume:medium - аналогично volume:50, если правило не установлено, то это значение по умолчанию.

В CSS есть целый ряд правил, который поможет правильно обработать ваши файлы

- **volume:loud** - аналогично volume:75.
  volume:x-loud - аналогично volume:100.
  volume:inherit - значение вычисленное браузером.
  volume:50% - громкость можно задавать в процентах, относительно наследуемого значения, а затем выравниваются в пределах от 0 до 100.
  Правило volume наследуется и используется только для звуковых носителей.

Правило speak: определяет будет ли текст произноситься, если да то как.

- Допустимые значения:
  speak:normal - значение по умолчанию, текст воспроизводится согласно языку.
  speak:none - текст не воспроизводится, как будто отсуствует.
  speak:spell-out - текст воспроизводится по буквам.
  speak:inherit - вычисленное браузером.

Правило pause-before: определяет паузу (задержку) перед воспроизведением бокса и имеет значения:

pause-before:700ms - пауза задается в секундах или милисекундах
pause-before:75% - задается в процентах относительно правила speech-rate, т.е. если speech-rate установлено 80 слов в минуту, а абзац содержит 120 слов то pause-before:100% бутет иметь смысл как pause-before:1500ms.
pause-before:inherit - значение вычисленное браузером.
Не наследуется и относится только к звуковым носителям.

Правило pause-after:

- определяет паузу после воспроизведения бокса и имеет те же значения что и pause-before

Правило pause: помещает в себе два предыдущих.

- **P{pause:100ms}** если указано одно значение то оно распространяется на 2 правила(pause-before и pause-after).
  DIV{pause:100ms 150ms} если указано два значения то первое распространяется на pause-before, а второе на pause-after.

Правило pause не наследуется.

Правило cue-before: определяет иконку(звуковой файл) перед боксом.

- Т.е. перед боксов будет проигрываться какая-то мелодия, возможно с словами. Правило имеет значения:
- **cue-before:none** значение по умолчанию, иконка отсутствует.
  cue-before:url("file.wav") абсолютный или относительный адрес иконки.
  cue-before:inherit - вычесленное браузером, если в браузере вмонтированы звуки, то он может подставить свои.
  Правило не наследуется и используется только для звуковых носителей

- Правило **cue-after:** определяет иконку (звуковой файл) после бокса.

Значения как и у cue-before

- Правило **cue:** - это сокращенная форма для cue-before и cue-after.**DIV{cue:url("file.wav")}**, может принимать два значения для начала и конца.

Правило play-during: устанавливает и управляет фоновой музыкой, т.е. каскадные таблицы стилей позволяют добавлять мелодию в время воспроизведения страницы.

- **Правило play-during имеет значения:**
  {play-during:url("file.wav")} - указывает путь к фоновой музыке.
  {play-during:url("file.wav") mix } - фоновая музыка в дескрипторе url смешивается с фоновой музыкой родительского элемента.
  {play-during:url("file.wav") repeat } - если время воспроизведение фоновой музыки короче времени воспроизведения страницы, то она повторяется.
  {play-during:auto} - это значение по умолчанию, которое имеет смысл что фоновая музыка родительского элемента воспроизводится без рестарта.
  {play-during:none } - фоновая музыка не воспроизводится.
- ***Правило play-during применяется только для звуковых носителей, оно не наследуется и его можно использовать с всеми элементами.***

- Правило **azimuth:** управляет звуковыми эффектами, по умолчанию воспроизводится моно, но можно использовать стерео, квадро или даже систему домашнего кинотеатра.
- Управление звуковой системы производится при помощи единиц измерения **deg** в диапазоне от -360 до 360,
- центр(перед) это 0deg,
- сзади = 180deg,
- слева -90(270)deg и справа 90deg.

Синтаксис правила azimuth:

- ***{azimuth:90deg -90deg}**
  Значения могут быть: left-side, far-left, left, center-left, center, center-right, right, far-right, right-side,,behind, leftwards, rightwards
- ***что они значат смотрите в спецификации.***

- Правило **elevation:** управляет звуковым эффектом по вертикали, некоторые системы(например ПсевдоСтерео) позволяет звуку литься сверху или снизу.

- Правило **elevation** так-же измеряется в единицах измерения **deg**, но в диапазоне от -90 до 90, -90deg звук льется по низу, 90deg звук идет по верху.

- Правило elevation может иметь только одно значение: **elevation:0deg** - задается в единицах измерения.
  elevation:below - тоже что и -90.
  elevation:level - значение по умолчанию, тоже что и ноль.
  elevation:above - тоже что и 90.
  elevation:higher - добаляет 10deg к значению родителя.
  elevation:lower - убаляет 10deg у значения родителя.
  Правила azimuth и elevation наследуются, применяются к всем элементам для звуковых типов носителей.

Свойства характеристики голоса

Правило speech-rate: управляет темпом речи, т.е. сколько слов в минуту должно воспроизводиться.

- Данное правило имеет значение:
  speech-rate:120 - значение задается цифрой.
  speech-rate:x-slow - 80 слов в минуту.
  speech-rate:slow - 120 слов в минуту.
  speech-rate:medium - значение по умолчанию, 180-200 слов в минуту.
  speech-rate:fast - 300 слов в минуту.
  speech-rate:x-fast - 500 слов в минуту.
  speech-rate: - ускорить на 40 слов в минуту.
  speech-rate: - замедлить на 40 слов в минуту.
  При выборе темпа нужно учитывать язык текста. Правило speech-rate наследуется, применяется к всем элементам и используется только к звуковым носителям

Правило voice-family: управляет стилем воспроизведения текста.

- **voice-family:male** - текст воспроизводится мужским голосом.
  voice-family:female - текст воспроизводится женским голосом.
  voice-family:child - текст воспроизводится детским голосом.
  Значения могут быть и специфичными например comedian, trinoids, carlos или lani. Есть и другие специфичные значения.
  Правило voice-family наследуется, применяется к всем элементам и используется только к звуковым носителям.

Правило pitch: управляет высотой(частотой) голоса и имеет значения:

- **pitch:150Hz** - высота голоса задается в герцах
  pitch:x-low - очень низкий голос.
  pitch:low - низкий голос.
  pitch:medium - средний голос.
  pitch:high - высокий голос.
  pitch:x-high - очень высокий голос.
  Правило pitch если оно имеет значение не в герцах, то оно только дополняет voice-family.

- Правило **pitch-range:** управляет оживлённостью голоса, задается числами:
  ***{pitch-range:50;}** -это начальное значение, его можно изменить от от 0 до 100.

Правило stress:

- управляет интонацией голоса, например по умолчанию stress:50, но если вам нужно подавить интонацию вы можете понизить stress:0 или повысить до stress:100.
- ***Это правило действует только вместе с pitch-range.***

Правило richness:

- управляет полетностью голоса, например на улице или в большем зале нужен более "полетнй" голос(richness:100), а в небольшой комнате нужен более мелодичный(мягкий) тогда richness:0.
- По умолчанию значение 50, не обязательно чтобы повысить или понизить использовать значение 0 или 100, вы можете самостоятельно определить это значение от 0 до 100, например 30 или 79.

- Правило **speak-punctuation:** управляет произношением знаков пунктуации и имеет значения:
  speak-punctuation:code - знаки пунктуации произносятся буквально, т.е. если в тексте встречается точка, то браузер произносит: "точка", тоже с запятымы, скобками кавычками.
  speak-punctuation:none - знаки пунктуации не произносятся.

- Правило **speak-numeral:** управляет произношением цифр и имеет следующие значения:
  speak-numeral:digits - цифры произносятся посимвольно, например число 123 браузер произнесет как: "один два три".
  speak-numeral:continuous - число произносится целиком, например число 123 браузер произнесет как: "сто двадцать три".

- **Патчи в какскадных таблиц стилей для I.E., FireFox, Opera и Safari**

CSS позволяет добиться межбраузерности

- т.е. дизайн будет отображаться в всех браузерах не зависимо от операционной системы одинаково, но это немного не так, что бы добиться абсолютной межбраузерностью нужно пользоваться "хаками" или проще говоря патчами.

- Патч от английского patch - это заплатка, которая покрывает какую либо дыру. В каскадных таблицах стилей есть тоже "дыры", т.е. некоторые браузеры неправильно обрабатывают документы, в элементарных правилах дыры не образовываются, а вот с свойствами правил таких как: display, position, float и превдоклассы начинаются проблемы. Например InternetExplorer некорректно понимает float, а I.E-6 не понимает :hover вместо него можно использовать :focus, и таких примеров может быть сотни..

Патчи для браузера FireFox

- для браузера FireFox всех версий:

```
@-moz-document url-prefix() {
.style {color:#000000;}
}
```

Патчи для браузера Opera

- Только для Оперы-9.5:

```
html:first-child .style {
color:#000000;
}
```

Для других версий оперы:

```
@media all and (-webkit-min-device-pixel-ratio:10000),
not all and (-webkit-min-device-pixel-ratio:0) {
.style {color:#000000;}
}
```

или

```
*|html[xmlns*=""] .style {
color:#000000;;
}
```

CSS-патчи для браузера Safari

- Первый вариант:
  body:last-child:not(:root:root) .style {
  color:#000000;
  }
  Второй вариант:
  html[xmlns*=""] body:last-child .style {
  color:#000000;
  }

CSS-патчи для браузера Internet Explorer

Для браузера Internet Explorer 6 :

```
.style {
_color:#000000;
}
```

или

```
* html .style {
background: #F00;
}
```

Для браузера Internet Explorer 7 :

```
*+html .style {
color:#000000;
}
```

или

```
*:first-child+html .style {
color:#000000;
}
```

- или

  ```
  html>body .style {
  color:#000000;
  }
  ```

- Патч для браузера Internet Explorer 8 :

  ```
  @media \0screen {
  .style {color:#000000;}
  ```

- Патч для браузера Internet Explorer 9 :

  ```
  :root .style {
  color:#000000;\9;
  }
  ```

Из-за постоянных проблем в Internet Explorer, были разработаны даже условные конструкции в которые помещают стили тегом link или style:

- <!--[if IE]>
  <style type="text/css">
  html{overflow:auto;} body{color:#000000;}
  </style>
  <![endif]-->
  <!--[if IE]> ... <![endif]--> - для всех версий Internet Explorer
  <!--[if IE 5]> ... <![endif]--> - только для 5 версии.
  <!--[if IE 6]> ... <![endif]--> - только для 6 версии.
  <!--[if IE 7]> ... <![endif]--> - только для 7 версии.
  <!--[if IE 8]> ... <![endif]--> - только для 8 версии.
  <!--[if IE 9]> ... <![endif]--> - только для 9 версии.
  <!--[if ! IE]> ... <![endif]--> - не для Internet Explorer.
  <!--[if lt IE 7]> ... <![endif]--> - для всех что ниже 7.
  <!--[if gt IE 6]> ... <![endif]--> - для всех что выше 6.
  <!--[if gte IE 7]> ... <![endif]--> - для 7 и выше.
  <!--[if lte IE 7]> ... <![endif]--> - для 7 и ниже.
  <!--[if ! gte IE 7]> ... <![endif]--> - для всех версий кроме 7.
  В последних 5 вариантах цифра может быть любая до 9.