

# Глава 4. Каскадные таблицы стилей

<b>Основные понятия.....</b>	<b>3</b>
Что такое каскадные таблицы стилей.....	3
Применение каскадных таблиц стилей.....	5
Блоковая структура форматирования.....	7
<b>Шрифты.....</b>	<b>8</b>
font-family.....	8
font-size.....	8
font-style.....	9
font-weight.....	9
line-height.....	9
font.....	9
face.....	10
letter-spacing.....	10
<b>Цвет и фон.....</b>	<b>10</b>
color.....	10
background-color.....	11
background.....	11
<b>Форматирование текста.....</b>	<b>11</b>
text-decoration.....	11
text-decoration-style.....	12
line-width.....	12
text-align.....	12
vertical-align.....	13
text-indent.....	13
white-space.....	13
<b>Таблицы.....</b>	<b>14</b>
table-layout.....	14
border-collapse.....	14
empty-cells.....	15
cellpadding.....	15
cellspacing.....	15
<b>Блоки.....</b>	<b>16</b>



<a href="#">margin-top.....</a>	<a href="#">16</a>
<a href="#">margin-right.....</a>	<a href="#">16</a>
<a href="#">margin-bottom.....</a>	<a href="#">16</a>
<a href="#">margin-left.....</a>	<a href="#">16</a>
<a href="#">margin.....</a>	<a href="#">17</a>
<a href="#">padding-top.....</a>	<a href="#">17</a>
<a href="#">padding-right.....</a>	<a href="#">17</a>
<a href="#">padding-bottom.....</a>	<a href="#">18</a>
<a href="#">padding-left.....</a>	<a href="#">18</a>
<a href="#">padding.....</a>	<a href="#">18</a>
<a href="#">border-top.....</a>	<a href="#">19</a>
<a href="#">border-right.....</a>	<a href="#">19</a>
<a href="#">border-bottom.....</a>	<a href="#">20</a>
<a href="#">border-left.....</a>	<a href="#">20</a>
<a href="#">border.....</a>	<a href="#">21</a>
<a href="#">display.....</a>	<a href="#">22</a>
<b><a href="#">Позиционирование.....</a></b>	<b><a href="#">22</a></b>
<a href="#">position.....</a>	<a href="#">22</a>
<a href="#">top.....</a>	<a href="#">22</a>
<a href="#">left.....</a>	<a href="#">23</a>
<a href="#">bottom.....</a>	<a href="#">23</a>
<a href="#">right.....</a>	<a href="#">23</a>
<a href="#">width.....</a>	<a href="#">24</a>
<a href="#">height.....</a>	<a href="#">24</a>
<a href="#">thead-position.....</a>	<a href="#">24</a>
<a href="#">tfoot-position.....</a>	<a href="#">25</a>
<b><a href="#">Разбивка на страницы и печать.....</a></b>	<b><a href="#">25</a></b>
<a href="#">size.....</a>	<a href="#">25</a>
<a href="#">@page.....</a>	<a href="#">26</a>
<a href="#">page-break-after.....</a>	<a href="#">26</a>
<a href="#">page-break-before.....</a>	<a href="#">26</a>
<a href="#">page-break-inside.....</a>	<a href="#">27</a>
<a href="#">couple-next.....</a>	<a href="#">27</a>
<a href="#">col-breaks.....</a>	<a href="#">27</a>
<b><a href="#">Вывод значений переменных.....</a></b>	<b><a href="#">28</a></b>
<a href="#">precision.....</a>	<a href="#">28</a>
<a href="#">zero.....</a>	<a href="#">28</a>
<a href="#">zero-padding.....</a>	<a href="#">29</a>



<a href="#">split-nums.....</a>	<a href="#">29</a>
<a href="#">negative.....</a>	<a href="#">29</a>
<a href="#">negative-color.....</a>	<a href="#">30</a>
<a href="#">in-words.....</a>	<a href="#">30</a>
<a href="#">point.....</a>	<a href="#">30</a>
<a href="#">num-width.....</a>	<a href="#">30</a>
<a href="#">date-formats.....</a>	<a href="#">31</a>
<a href="#">_enabled.....</a>	<a href="#">31</a>
<a href="#">@media print.....</a>	<a href="#">31</a>
<a href="#">onprint.....</a>	<a href="#">32</a>
<b><a href="#">Прочее.....</a></b>	<b><a href="#">32</a></b>
<a href="#">_cursor.....</a>	<a href="#">32</a>

## Основные понятия

### Что такое каскадные таблицы стилей

Каскадные таблицы стилей – это технология определения стилей, которые влияют на форматирование тегов в отчётной форме. Аналог таблицам стилей можно найти в стилях для документа Microsoft Word. Слово «каскадные» в названии означает, что приоритетность при возникающих конфликтах определяется при выстраивании всех правил каскадом, т. е. наибольший приоритет имеет описанное последним правило.

Каждое правило форматирования состоит из двух частей: селектора и определения.

Селектором может быть любой тег, для которого в определении задается способ его форматирования. Такое правило влияет на отображение всех элементов, выведенных данным тегом. Если есть необходимость в определении одного и того же стиля для нескольких тегов, то их можно записать через запятую в один селектор. Так же помощью параметра Class можно задать разные правила форматирования для одного элемента данного типа или всех элементов документа. Имя класса указывается в селекторе после имени тега через точку. Имя класса в параметре указывается без точки и может быть заключено в кавычки или нет. Если класс должен применяться к любым элементам отчётной формы, то в селекторе не указывается имя тега, а пишется точка и имя класса.

Каждое определение состоит из двух частей: свойства и его значения. Эти части отделяются друг от друга двоеточием. Значение свойства определяется из его названия. Все свойства можно разделить на несколько категорий, каждой из которых далее будет посвящен свой раздел.

В одном правиле возможно задавать несколько определений, разделенных символом «;». Селекторы, для которых задается одинаковое правило, тоже можно сгруппировать, записав через «,».

Если вложенный тег не имеет своего правила форматирования, то он унаследует его от своего элемента-родителя. Наследование полезно при задании значений свойств, используемых по умолчанию. Для этого достаточно задать свойства для элемента, порождающего все остальные элементы отчётной формы.

Назначить элементу отчётной формы стилевые описания достаточно просто. Нужно в качестве селектора указать сам элемент и после записать всю информацию о его форматировании, заключив ее в фигурные скобки. Например, для того, чтобы задать по умолчанию отображение текста шрифтом **Times New Roman** черного цвета, следует в спецификации записать:

```
body { color: black;
       font-family: Times New Roman; }
```

Рассмотрим еще один пример стилевых описаний с использованием классов и групп селекторов. Определим по умолчанию черный шрифт Arial и выравнивание по левому краю, а для заголовков первого и второго уровня – шрифт семейства **Tahoma** и выравнивание по центру. Заголовок первого уровня должен выводиться шрифтом размера **18pt**. Зададим формат заголовка второго уровня и возможность выводить его курсивом. Так же определим один абстрактный класс **textborder**, с помощью которого будет выводиться текст в рамке.

```
body { color: black;
       text-align: left;
       font-family: Arial;
       font-size: 10pt; }
h1, h2 { font-family: Tahoma;
        text-align: center;
        color: red; }
h1 { font-size: 18pt; }
h2 { font-size: 16pt; }
h2.italic { font: bold italic; }
.textborder { border: 1px solid black; }
```

Ниже один из вариантов применения определенных выше стилей в теле отчётной формы:

```
<body>
  <h1>заголовок первого уровня</h1><br>
  <h2>заголовок второго уровня</h2><br>
  <h2 class="italic">заголовок второго уровня
курсивом</h2><br>
  <p>просто текст (по умолчанию)</p>
  <p class="textborder">текст в рамке</p>
</body>
```

## Применение каскадных таблиц стилей

Каскадные таблицы стилей имеют множество преимуществ. По возможности стоит разделять структуру отчётной формы и оформление отчёта.

Язык разметки является основой для создания отчётных форм и таблицы стилей не заменяют его. Они являются самостоятельной технологией, которая применяется непосредственно к тегу.

Информацию о стиле элементов можно включить в отчётную форму тремя способами:

- связать отчётную форму с внешней таблицей стилей;
- внедрить глобальную таблицу стилей в заголовок отчётной формы;
- определить внутренний стиль для отдельного элемента.

## Внешняя таблица стилей

Все правила форматирования элементов отчётной формы могут быть вынесены в отдельный файл с расширением **.css**. Файлы с описаниями стилевых описаний могут располагаться где угодно. В отчётную форму внешний файл с описанием таблиц стилей присоединяется с помощью тега [<link>](#), который располагается в разделе [<head>](#). В качестве параметра **rel** тега [<link>](#) нужно указать значение **stylesheet**, в параметре **type** – «text/css», а в параметре **href** – полный путь к стилевому файлу.

```
<head>
  <link rel="stylesheet" type="text/css"
href="C:\СБиС++ Хозрасчетный
учет\Модули\!sbis\Отчёты\my.css">
  <title>
    заголовок отчёта
  </title>
```

</head>

Пример внешней таблицы стилей можно посмотреть в папке <Модули\!sbis\main.css>. Этот файл автоматически присоединяется программой СБиС++ к любой создаваемой в ней отчётной форме.

Оформление большинства отчётов должно быть одинаковым или схожим. Разумно разработать единую для всех отчётов стилевую спецификацию, вынести ее во внешний файл и каждый раз использовать при построении отчётных форм. Это упростит задачу поддержания общего стиля и облегчит обновление отчётных форм.

## Внедрение таблицы стилей в заголовок

Стилевые спецификации можно записать непосредственно в отчётной форме. Для этого нужно в разделе **head** поместить элемент `<style>`, а внутри пары `<style>` и `</style>` записать все правила форматирования.

Можно размещать несколько элементов `<style>` в разделе **head** одной отчётной формы. Так же возможно сочетание внешней таблицы стилей и внедрения таблиц в текст отчётной формы.

Ниже приведен пример внедрения описания стилей в раздел **head** отчётной формы:

```
<head>
  <style>
    body { font-family: arial;
           text-align: left; }
    h1 { font-size: 20pt;
         text-align: center;
         color: red; }
  </style>
  <title>
    заголовок отчёта
  </title>
</head>
```

## Внутренние стили отдельных элементов

Третьим вариантом добавления стилевых таблиц является применение их к конкретному элементу отчётной формы. Для этих целей у большинства тегов существует параметр `style`. Правила форматирования указываются в кавычках в качестве значения этого параметра. Рассмотрим пример определения внутреннего стиля для элемента `<h1>`:

```
<h1 style="font-size: 20pt; text-align: center;
color: red;"> заголовок первого уровня </h1>
```

## Блоковая структура форматирования

В каскадных таблицах используется модель представления документа в виде блоков. Прямоугольные блоки определяются для элементов отчётной формы и отображаются в нем в соответствии с моделью визуального форматирования.

Каждый блок имеет информативную область, в которой заключено содержимое элемента (например, текст, изображение и т.п.). Кроме того, каждый блок может также иметь области, отведенные для оформления отступов, границ и полей. Размер каждой из этих областей задается свойствами, описанными ниже. Рисунок 1 иллюстрирует взаимное расположение полей, границ и отступов.

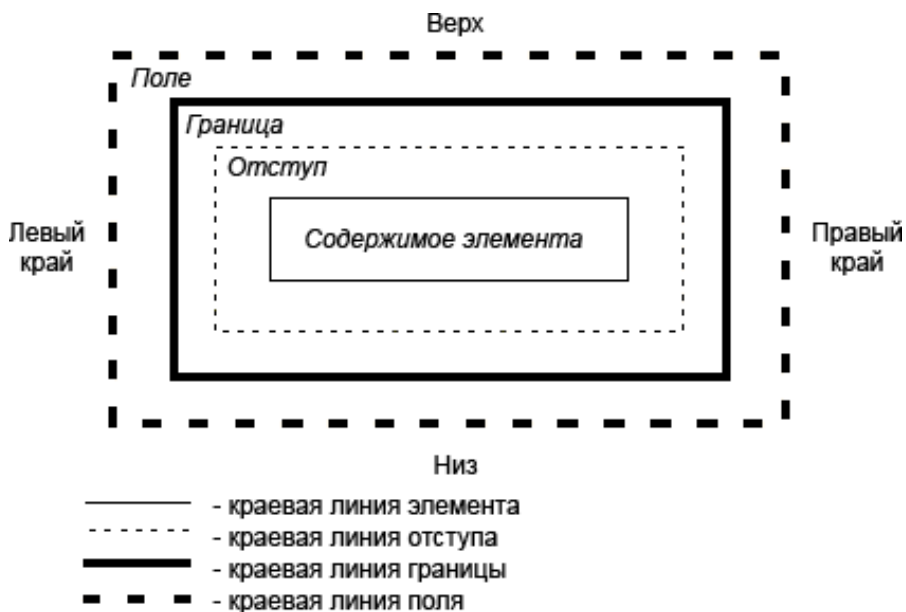


Рис. 1. Блоковая модель форматирования элемента и ее параметры

Краевая линия элемента ограничивает отображаемое содержимое элемента, порождающего блок. Далее идет область отступа, которая всегда имеет цвет фона совпадающий с фоном самого элемента. Если эта область имеет нулевую ширину, то край отступов совпадает с краем элемента. Область границы отделяет элемент с отступом от поля. Если ширина границы равна нулю, то ее край совпадает с краем отступов. Самой внешней областью является поле. Поле всегда является прозрачным и если ширина поля равна нулю, то ее край совпадает с краем границы.

Свойства таблиц стилей позволяют устанавливать различные размеры и цвета отдельных областей. Размеры всего блока складываются из размеров самого элемента и размеров отступа, границы и поля.

Каждая краевая линия может быть разбита на четыре сегмента: верхний, нижний, левый и правый, каждый из которых называют соответствующим краем. По такому же принципу разбиваются и сами области.

*Ширина* элемента (**width**) определяется расстоянием между левым и правым краем всего блока. *Высота* элемента (**height**) – расстояние между верхним и нижним краем.

## Шрифты

### font-family

Позволяет задавать приоритетный список шрифтов или семейств шрифтов.

#### Синтаксис

```
font-family : Шрифт1, Шрифт2, ...
```

#### Значения

**Шрифт1, Шрифт2, ...** - название шрифтов или семейств шрифтов. При построении отчётной формы браузер в первую очередь попытается использовать шрифт1. Если такой шрифт отсутствует, то браузер попытается применить шрифт2 и т.д. Семейство шрифтов вводится в случае, если не найдены все предшествующие варианты. Обычно на большинстве компьютеров есть основные пять типов семейств шрифтов: **serif, sans-serif, cursive, fantasy, monospace**.

### font-size

Задаёт размер шрифта.

#### Синтаксис

```
font-size : Значение
```

#### Значения

**Число**, задающее размер шрифта, и соответствующие [единицы измерения](#).



## font-style

Определяет стиль шрифта указанного семейства.

### Синтаксис

`font-style` : **Значение**

### Значения

**Normal** - обычный прямой шрифт.

**Italic** - курсивный шрифт, близок к рукописному.

**None** - устанавливает значение по умолчанию, т.е. normal.

## font-weight

Позволяет выбрать шрифт из семейства заданной жирности.

### Синтаксис

`font-weight` : **Значение**

### Значения

**Bold** - полужирный шрифт.

**None** - устанавливает шрифт нормальной толщины.

## line-height

Задаёт высоту строки.

### Синтаксис

`line-height` : **Значение**

### Значения

**Число**, задающее высоту строки, и соответствующие [единицы измерения](#). Если заданное число меньше высоты текста, то высота строки будет полагаться равной высоте шрифта.

## font

Устанавливает в одном определении значения свойств **font-style**, **font-weight**, **font-size**, **font-family** и **line-height**.

## Синтаксис

`font` : **Стиль Жирность Размер Интервал Шрифт**

## Значения

Все значения перечисленных свойств задают через пробелы в указанном порядке. Значения некоторых свойств могут быть опущены. В этом случае значениям этих свойств будет присвоено значение `normal`.

## face

Аналог свойства `font-family` для тега `font`.

## Синтаксис

`face` : **Шрифт1, Шрифт2, ...**

## Значения

**Шрифт1, Шрифт2, ...** - название шрифтов или семейств шрифтов. При построении отчётной формы браузер в первую очередь попытается использовать шрифт1. Если такой шрифт отсутствует, то браузер попытается применить шрифт2 и т.д. Семейство шрифтов вводится в случае, если не найдены все предшествующие варианты. Обычно на большинстве компьютеров есть основные пять типов семейств шрифтов: **serif, sans-serif, cursive, fantasy, monospace**.

## letter-spacing

Позволяет задать расстояние между соседними символами строки.

## Синтаксис

`letter-spacing` : **значение**

## Значения

**Число**, задающее расстояние между элементами строки, и соответствующие [единицы измерения](#).

## Цвет и фон

### color

Устанавливает цвет текста.



### **Синтаксис**

`color` : значение

### **Значения**

Зарезервированное название цвета или его значение в палитре RGB.

## **background-color**

Устанавливает цвет фона.

### **Синтаксис**

`background-color` : значение

### **Значения**

Зарезервированное название цвета или его значение в палитре RGB.

## **background**

Позволяет установить прозрачность.

### **Синтаксис**

`background` : значение

### **Значения**

`Transparent` - прозрачный фон.

`None` - непрозрачный, принимается по умолчанию.

## **Форматирование текста**

### **text-decoration**

Позволяет подчеркнуть, зачеркнуть и начертить линию над текстом.

### **Синтаксис**

`text-decoration` : значение1[ значение2[ значение3]]

### **Значения**

`Underline` - подчеркивание.

**Overline** - надчеркивание.

**Line-through** - зачеркивание.

## Комментарии

Все три свойства могут быть применены вместе или попарно. При этом нужные значения необходимо записать через пробел

## text-decoration-style

Задаёт тип линии для свойства text-decoration.

### Синтаксис

```
text-decorated : значение
```

### Значения

**None** - линия не отображается.

**Solid** - сплошная линия.

**Double** - двойная сплошная линия.

**Dashed** - пунктирная линия.

**Dotted** - точечная линия.

**DashDot** - тире-точка.

**DashDotDot** - тире-точка-точка.

## line-width

Устанавливает область действия свойства text-decoration.

### Синтаксис

```
line-width : значение
```

### Значения

**Width** - свойство text-decoration распространяется до конца блока.

**Short** - свойство text-decoration применяется только к части блока.

## text-align

Задаёт выравнивание элемента в блоке по горизонтали.



## Синтаксис

`text-align` : значение

## Значения

**Left** - выравнивание по левой границе блока отображения отчётной формы.

**Center** - выравнивание по центру блока отображения отчётной формы.

**Right** - выравнивание по правой границе блока отображения отчётной формы.

**None** - использовать значение по умолчанию, т.е. Left.

## vertical-align

Задает выравнивание элемента в блоке по вертикали.

## Синтаксис

`vertical-align` : значение

## Значения

**Top** - выравнивание по верхнему краю.

**Middle** - выравнивание по середине.

**Bottom** - выравнивание по нижнему краю.

**None** - не выравнивать.

## text-indent

Определяет отступ первой строки текста в блоке.

## Синтаксис

`text-indent` : значение

## Значения

**Число**, задающее расстояние отступа, и соответствующие [единицы измерения](#). По умолчанию отступ полагается равным 0.

## white-space

Определяет способ форматирования большого текста внутри элемента.

### Синтаксис

`white-space` : значение

### Значения

**Normal** - автоматическая разбивка текста на строки, если он не вмещается целиком в блок.

**Nowrap** - разбивка текста на строки происходит только с помощью тега `<br>`.

## Таблицы

### `table-layout`

Определяет способ размещения ячеек, строк и столбцов таблицы.

### Синтаксис

`table-layout` : значение

### Значения

**Fixed** - размер таблицы фиксируется и не изменяется.

**Auto** - размер таблицы может изменяться в зависимости от ее содержимого.

### `border-collapse`

Задаёт тип наложения границ ячеек в таблице.

### Синтаксис

`border-collapse` : значение

### Значения

**Collapse** - границы соседних ячеек находят одна на другую.

**Separated** - у каждой ячейки своя граница, между которыми расстояние равно ширине поля.

**None** - устанавливает значение по умолчанию, т.е. collapse.



## empty-cells

Задаёт отображение пустых ячеек.

### Синтаксис

`empty-cells` : значение

### Значения

**Show** - прорисовывать границы вокруг пустых ячеек.

**Hide** - не отображать пустые ячейки.

## cellpadding

Определяет расстояние между рамкой ячейки и данными внутри ячейки.

### Синтаксис

`cellpadding` : значение

### Значения

**Число**, задающее расстояние от рамки до данных ячейки, и соответствующие [единицы измерения](#). По умолчанию принимается значение 1px.

## cellspacing

Определяет расстояние между смежными ячейками по горизонтали и вертикали.

### Синтаксис

`cellspacing` : значение

### Значения

**Число**, задающее расстояние между ячейками, и соответствующие [единицы измерения](#). По умолчанию значение равно 2px. При задании значения 0 создается впечатление единой сетки таблицы.

## Блоки

### margin-top

Задаёт ширину верхнего поля в блоке.

#### Синтаксис

`margin-top` : значение

#### Значения

Число, задающее ширину поля, и соответствующие [единицы измерения](#). По умолчанию значения всех полей равны 0.

### margin-right

Задаёт ширину правого поля в блоке.

#### Синтаксис

`margin-right` : значение

#### Значения

Число, задающее ширину поля, и соответствующие [единицы измерения](#). По умолчанию значения всех полей равны 0.

### margin-bottom

Задаёт ширину нижнего поля в блоке.

#### Синтаксис

`margin-bottom` : значение

#### Значения

Число, задающее ширину поля, и соответствующие [единицы измерения](#). По умолчанию значения всех полей равны 0.

### margin-left

Задаёт ширину левого поля в блоке.



## Синтаксис

`margin-left` : значение

## Значения

Число, задающее ширину поля, и соответствующие [единицы измерения](#). По умолчанию значения всех полей равны 0.

## margin

Устанавливает значение верхнего, нижнего, правого и левого поля.

## Синтаксис

`margin` : `верхнеполе` `правополе` `нижнеполе` `левополе`

## Значения

Числа, задающие значения всех свойств, и соответствующие [единицы измерения](#), записанные через пробел. Если заданы одно, два или три значения, то недостающие свойства полагаются равными последнему значению.

## padding-top

Задаёт ширину верхнего отступа в блоке.

## Синтаксис

`padding-top` : значение

## Значения

Число, задающее ширину отступа, и соответствующие [единицы измерения](#). По умолчанию значения всех отступов равны 0.

## padding-right

Задаёт ширину правого отступа в блоке.

## Синтаксис

`padding-right` : значение



## Значения

Число, задающее ширину отступа, и соответствующие [единицы измерения](#). По умолчанию значения всех отступов равны 0.

## padding-bottom

Задаёт ширину нижнего отступа в блоке.

### Синтаксис

```
padding-bottom : значение
```

## Значения

Число, задающее ширину отступа, и соответствующие [единицы измерения](#). По умолчанию значения всех отступов равны 0.

## padding-left

Задаёт ширину левого отступа в блоке.

### Синтаксис

```
padding-left : значение
```

## Значения

Число, задающее ширину отступа, и соответствующие [единицы измерения](#). По умолчанию значения всех отступов равны 0.

## padding

Устанавливает значение верхнего, нижнего, правого и левого отступа.

### Синтаксис

```
padding : верхотступ правотступ нижотступ левотступ
```

## Значения

Числа, задающие значения всех свойств, и соответствующие [единицы измерения](#), записанные через пробел. Если заданы одно, два или три значения, то недостающие свойства полагаются равными последнему значению.



## border-top

Задаёт толщину, тип линии и цвет верхней границы в блоке.

### Синтаксис

`border-top` : толщина типлинии цвет

### Значения

Значения всех свойств через пробел. Для задания значения толщины необходимо указать число и соответствующие [единицы измерения](#). По умолчанию толщина равна 0. Тип линии задается ключевым словом. Доступны следующие типы линий:

**None** - граница не отображается.

**Solid** - сплошная линия.

**Double** - двойная сплошная линия. При этом сумма толщины двух линий и промежутка между ними равна толщине.

**Dashed** - пунктирная линия.

**Dotted** - точечная линия.

**DashDot** - тире-точка.

**DashDotDot** - тире-точка-точка.

Для цвета используется либо его зарезервированное название, либо его значение в палитре RGB.

## border-right

Задаёт толщину, тип линии и цвет правой границы в блоке.

### Синтаксис

`border-right` : толщина типлинии цвет

### Значения

Значения всех свойств через пробел. Для задания значения толщины необходимо указать число и соответствующие [единицы измерения](#). По умолчанию толщина равна 0. Тип линии задается ключевым словом. Доступны следующие типы линий:

**None** - граница не отображается.

**Solid** - сплошная линия.



**Double** - двойная сплошная линия. При этом сумма толщины двух линий и промежутка между ними равна толщине.

**Dashed** - пунктирная линия.

**Dotted** - точечная линия.

**DashDot** - тире-точка.

**DashDotDot** - тире-точка-точка.

Для цвета используется либо его зарезервированное название, либо его значение в палитре RGB.

### **border-bottom**

Задаёт толщину, тип линии и цвет нижней границы в блоке.

#### **Синтаксис**

`border-bottom` : **толщина типлинии цвет**

#### **Значения**

Значения всех свойств через пробел. Для задания значения толщины необходимо указать число и соответствующие [единицы измерения](#). По умолчанию толщина равна 0. Тип линии задается ключевым словом. Доступны следующие типы линий:

**None** - граница не отображается.

**Solid** - сплошная линия.

**Double** - двойная сплошная линия. При этом сумма толщины двух линий и промежутка между ними равна толщине.

**Dashed** - пунктирная линия.

**Dotted** - точечная линия.

**DashDot** - тире-точка.

**DashDotDot** - тире-точка-точка.

Для цвета используется либо его зарезервированное название, либо его значение в палитре RGB.

### **border-left**

Задаёт толщину, тип линии и цвет левой границы в блоке.



## Синтаксис

`border-left` : толщина типлинии цвет

## Значения

Значения всех свойств через пробел. Для задания значения толщины необходимо указать число и соответствующие [единицы измерения](#). По умолчанию толщина равна 0. Тип линии задается ключевым словом. Доступны следующие типы линий:

**None** - граница не отображается.

**Solid** - сплошная линия.

**Double** - двойная сплошная линия. При этом сумма толщины двух линий и промежутка между ними равна толщине.

**Dashed** - пунктирная линия.

**Dotted** - точечная линия.

**DashDot** - тире-точка.

**DashDotDot** - тире-точка-точка.

Для цвета используется либо его зарезервированное название, либо его значение в палитре RGB.

## border

Задаёт толщину, тип линии и цвет для всей границы в блоке.

## Синтаксис

`border` : толщина типлинии цвет

## Значения

Значения всех свойств через пробел. Для задания значения толщины необходимо указать число и соответствующие [единицы измерения](#). По умолчанию толщина равна 0. Тип линии задается ключевым словом. Доступны следующие типы линий:

**None** - граница не отображается.

**Solid** - сплошная линия.

**Double** - двойная сплошная линия. При этом сумма толщины двух линий и промежутка между ними равна толщине.

**Dashed** - пунктирная линия.

**Dotted** - точечная линия.

**DashDot** - тире-точка.

**DashDotDot** - тире-точка-точка.

Для цвета используется либо его зарезервированное название, либо его значение в палитре RGB.

## display

Задаёт способ отражения блоков при создании элемента.

### Синтаксис

`display` : значение

### Значения

**Block** - элемент порождает один структурный блок.

**Inline** - элемент порождает один или несколько строковых блоков.

**None** - элемент не порождает ни одного блока.

## Позиционирование

### position

Определяет положение блока в отчётной форме.

### Синтаксис

`position` : значение

### Значения

**Static** - блок позиционируется в соответствии с предыдущими элементами.

**Relative** - блок смещается относительно предыдущих с помощью свойств **left**, **right**, **top** и **bottom**.

**Absolute** - положение блока указывается с помощью свойств **left**, **right**, **top** и **bottom**.

### top

Задаёт смещение верхней краевой линии поля относительно положения блока (свойство **position** установлено в **relative**) или краев документа (свойство **position** равно **absolute**).



## Синтаксис

`top` : значение

## Значения

Число, задающее расстояние на которое смещается элемент, и соответствующие [единицы измерения](#).

## left

Задаёт смещение левой краевой линии поля относительно положения блока (свойство **position** установлено в **relative**) или краев документа (свойство **position** равно **absolute**).

## Синтаксис

`left` : значение

## Значения

Число, задающее расстояние на которое смещается элемент, и соответствующие [единицы измерения](#).

## bottom

Задаёт смещение нижней краевой линии поля относительно положения блока (свойство **position** установлено в **relative**) или краев документа (свойство **position** равно **absolute**).

## Синтаксис

`bottom` : значение

## Значения

Число, задающее расстояние на которое смещается элемент, и соответствующие [единицы измерения](#).

## right

Задаёт смещение правой краевой линии поля относительно положения блока (свойство **position** установлено в **relative**) или краев документа (свойство **position** равно **absolute**).

## Синтаксис

`right` : значение

## Значения

Число, задающее расстояние на которое смещается элемент, и соответствующие [единицы измерения](#).

## width

Задаёт ширину содержимого блока.

## Синтаксис

`width` : значение

## Значения

Число, задающее ширину элемента, и соответствующие [единицы измерения](#).

## height

Задаёт высоту содержимого блока.

## Синтаксис

`height` : значение

## Значения

Число, задающее высоту элемента, и соответствующие [единицы измерения](#).

## thead-position

Определяет положение верхнего колонтитула в таблице.

## Синтаксис

`thead-position` : значение

## Значения

**None** - колонтитул выводится на каждой странице.

**Middle** - выводить на отдельной странице.



**Table** - только в начале всей таблицы.

**Left** - только в начале левых страниц.

**Right** - только в начале правых страниц.

## **tfoot-position**

Определяет положение нижнего колонтитула в таблице.

### **Синтаксис**

`tfoot-position` : значение

### **Значения**

**None** - колонтитул выводится на каждой странице.

**Middle** - выводить на отдельной странице.

**Table** - только в конце всей таблицы.

**Left** - только перед разрывом на левых страницах.

**Right** - только перед разрывом на правых страницах.

# **Разбивка на страницы и печать**

## **size**

Позволяет определить ориентацию страницы.

### **Синтаксис**

`size` : значение

### **Значения**

**Portrait** - вертикальная ориентация страницы, т.е. наиболее широкая сторона располагается вертикально.

**Landscape** - горизонтальная ориентация страницы, т.е. наиболее широкая сторона располагается горизонтально.

### **Комментарии**

Для тега `font` свойство `size` является аналогом свойства `font-size` и задает размер шрифта. Во всех остальных тегах соответствует вышеуказанному описанию.

## @page

Это отдельное правило, в котором устанавливаются размеры страницы и отступы.

### Синтаксис

```
@page {[size: ориентация;} [margin: отступы]}
@page {[size: ориентация;} [margin-top: верх;}
[margin-right: право;} [margin-bottom: низ;}
[margin-left: лево]}
```

### Значения

В определении правила в указанном порядке могут быть заданы ориентация страницы (в свойстве **size**) и отступы (свойства **margin** или **margin-top**, **margin-left**, **margin-bottom** и **margin-right**).

## page-break-after

Устанавливает принудительный разрыв страниц после блока.

### Синтаксис

```
page-break-after : значение
```

### Значения

**Auto** - не принуждает, но и не запрещает разрыв.

**Avoid** - запрещает разрыв страницы.

**Always** - всегда генерирует разрыв.

**Left** - генерирует один или два разрыва страниц, так что следующая страница форматируется как левая страница.

**Right** - генерирует один или два разрыва страниц, так что следующая страница форматируется как правая страница.

## page-break-before

Устанавливает принудительный разрыв страниц до блока.

### Синтаксис

```
page-break-before : значение
```



## Значения

**Auto** - не принуждает, но и не запрещает разрыв.

**Avoid** - запрещает разрыв страницы.

**Always** - всегда генерирует разрыв.

**Left** - генерирует один или два разрыва страниц, так что следующая страница форматируется как левая страница.

**Right** - генерирует один или два разрыва страниц, так что следующая страница форматируется как правая страница.

## page-break-inside

Устанавливает принудительный разрыв страниц внутри блока.

### Синтаксис

`page-break-inside` : значение

### Значения

**Auto** - не принуждает, но и не запрещает разрыв.

**Avoid** - запрещает разрыв страницы.

## couple-next

Позволяет выводить таблицу на одной странице со следующим за ней блоком.

### Синтаксис

`couple-next` : значение

### Значения

**On** - таблица выводится на одной странице со за ней следующим блоком при условии, что следующий блок занимает менее 1/3 страницы.

**Off | None** - следующий блок выводится произвольно.

## col-breaks

Позволяет переносить полностью столбец на следующую страницу, если таблица полностью не влезает по ширине страницы.

**Синтаксис**

`col-breaks` : значение

**Значения**

**On** - если столбец не влезает на страницу, то он полностью переносится на следующую.

**Off | None** - таблица разрывается в произвольном порядке, если она не влезает по ширине страницы целиком.

## Вывод значений переменных

**precision**

Устанавливает количество знаков, вводимых после запятой для отображения значений числовых переменных.

**Синтаксис**

`precision` : значение

**Значения**

Число, определяющее количество символов после запятой.

**zero**

Устанавливает возможность отображения заданного символа вместо 0 при выводе в отчётную форму значений числовых переменных.

**Синтаксис**

`zero` : значение

**Значения**

**Символ** - будет выводиться указанный символ.

**Empty** - ничего не будет выводиться.

**None** - будет выводиться как обычное число.

## zero-padding

Позволяет при выводе в отчётную форму значений числовых переменных не отображать лишние нули справа.

### Синтаксис

`zero-padding` : значение

### Значения

**On** - не отображать нули справа.

**Off** - выводить так как есть.

## split-nums

Определяет разбиение чисел на порядки при выводе в отчётную форму значений числовых переменных.

### Синтаксис

`split-nums` : значение

### Значения

**Символ** - будет выводиться указанный символ.

**On** - по умолчанию порядки отделяются с помощью пробела.

**Off** - разбиение не производится.

## negative

Определяет способ отображения отрицательных чисел при выводе в отчётную форму значений числовых переменных.

### Синтаксис

`negative` : значение

### Значения

**()** - отрицательные числа будут выводиться в скобках.

**None** - отменяет установленные ранее значения.

## negative-color

Определяет цвет отображения отрицательных чисел при выводе в отчётную форму значений числовых переменных.

### Синтаксис

```
negative-color : значение
```

### Значения

Зарезервированное название цвета или его значение в палитре RGB.

## in-words

Задаёт возможность отображения значений переменных прописью.

### Синтаксис

```
in-words : значение
```

### Значения

**On** - значения переменных выводятся прописью.

**Off** - отменяет установленные ранее значения.

## point

Позволяет установить символ для разделения целой и дробной части при отображении значений числовых переменных.

### Синтаксис

```
point : значение
```

### Значения

**Символ**, который будет выводиться для разделения целой и дробной части при отображении числа.

## num-width

Позволяет установить количество выводимых в отчётную форму символов при отображении значений числовых переменных.



## Синтаксис

`num-width` : значение

## Значения

Число, которое задает количество выводимых символов при отображении значений числовых переменных.

## Комментарии

Если символов в числе больше, чем задано, то старшие разряды обрезаются, и слева выводится символ «<». Если символов в числе меньше, то слева дописываются нули.

## date-formats

Задаёт маску для отображения значений переменных типа время.

## Синтаксис

`date-formats` : значение

## Значения

Текст - шаблон, описывающий порядок записи дня, месяца и года. В шаблоне допускается использовать следующие символы: "д" - день, "м" - номер месяца, "г" - год.

## enabled

Делает возможным редактирование поля ввода.

## Синтаксис

`enabled` : значение

## Значения

On - поле ввода может редактироваться пользователем.

Off | None - поле ввода не редактируемое.

## @media print

Это отдельное правило, которое позволяет задавать свойства элементов для вывода на печатающее устройство.

## Синтаксис

```
@media print {Правило}
```

## Значения

В фигурных скобках после названия указывается правило, которое должно быть применено для вывода указанного в нем элемента на печать.

## Комментарии

Это правило позволяет переопределять форматирование элементов определенного типа при выводе на печать. При этом к тому же самому типу тега может быть описано свое правило вне `media print`, которое будет отвечать за отображение элемента на экране.

## onprint

Устанавливает выполнение скрипта только перед печатью.

## Синтаксис

```
onprint : значение
```

## Значения

On - скрипт выполняется только перед печатью

Off | None - скрип выполняется в том месте, где он встречается.

# Прочее

## cursor

Задаёт вид курсора для блока.

## Синтаксис

```
cursor : значение
```

## Значения

Arrow - стандартная стрелка.

Circle - курсор, означающий запрет действий на объекте. Обычно представляется в виде перечеркнутого круга.

Crosshair - перекрестие, напоминающие символ «+».



**Default** - курсор, используемый на данной платформе по умолчанию. Обычно представляется в виде стрелки.

**Hand** - курсор, определяющий наличие действия на объекте.

**Help** - курсор, означающий, что для объекта, на который он указывает, имеется справочная информация. Зачастую представляется в виде вопросительного знака.

**Hourarrow** - курсор, указывающий на занятость программы, когда пользователю необходимо подождать. Обычно представляется в виде стрелочки с песочными часами.

**Hourglass** - курсор, указывающий на занятость программы, когда пользователю необходимо подождать. Обычно представляется в виде песочных часов.

**Nesw** - стрелочка из верхнего правого угла в нижний левый.

**Nwse** - стрелочка из нижнего левого в верхний правый угол.

**Northsouth** - курсор, указывающий на возможность переместить объект вверх-вниз.

**Size** - курсор, определяющий объект, у который можно изменить размер.

**Text** - курсор, используемый при выделении текста. Обычно представляется в виде вертикальной линии.

**Westeast** - курсор, указывающий на возможность переместить объект вправо-влево.