

CSS3-фильтры/ CSS3-медиазапросы/ CSS3 columns

Vira Huskova

CSS3-медиазапросы

В 2001 году в HTML4 и CSS2 была введена поддержка аппаратно-зависимых таблиц стилей, позволившая создавать стили и таблицы стилей для определенных типов устройств.

В качестве медиа-типов были определены следующие:

`aural`, `braille`, `handheld`, `print`, `projection`, `screen`, `tty`, `tv`.

Таким образом, браузер применял таблицу стилей только в случае, когда активизировался данный тип устройства.

Кроме того, было введено ключевое слово `all`, которое использовалось, чтобы указать, что таблица стилей применяется ко всем типам носителей.

В HTML4 медиа-запрос записывался следующим образом:

```
<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css">  
<link rel="stylesheet" type="text/css" media="print" href="serif.css">
```

Внутри таблицы стилей также можно было объявить, что блоки объявлений должны применяться к определенным типам носителей:

```
@media screen {  
    * {font-family: sans-serif;}  
}
```

Предусматривая возможность введения новых значений и значений с параметрами в будущем, для браузеров была реализована поддержка значений атрибута медиа-носителя.

Текущий синтаксис HTML5 и CSS3 напрямую ссылается на первую спецификацию Media Queries, обновляя правила для HTML. Также был расширен список характеристик медиа-носителей.

IE: 9.0 (кроме вложенных медиазапросов)

Edge: 12.0

Firefox: 3.5

Chrome: 26.0

Safari: 6.1

Opera: 10.1

iOS Safari: 7.1

Android: 4.4

Chrome for Android: 55.0

1

Established Nottingham 2003
THE CELEBRATED NEW MISCELLANY OF
MR. SIMON COLLISON
A.K.A. COLLY

Bested for your pleasure
POTTED AUTOBIOGRAPHY



Hello. I'm a freelance designer, speaker, and author based in Nottingham, England. I've written a few *books*, and I love doing *presentations* and workshops, plus occasional *interviews*. [Read More](#) →

Dropping science like it's hot
THE SPLENDID JOURNAL



S New Adventures. Last week, I announced New Adventures in Web Design, a unique and affordable web event that will take place in Nottingham, England. [A... More](#) →

Catalogued nocturnal matter
EXHAUSTIVE ARCHIVES



809—*Carry On Responsively*
808—*Gloucestershire Festival 2010*
807—*Memories of another trip*
806—*Thoughts on FOWD 2010*
805—*Design To Communicate*

Mr. Collison is currently
AVAILABLE FOR HIRE



Options & queries this way. Drop me a line if you wish. I'm currently considering all new projects, including design and development, writing, presentations, workshops, and consultation. →

EXTERNAL REFERENCES ([VIEW ALL](#))

Playing on the grasshopper
LAST.FM SCRIBBLES



Stashin' items from other folk
DELICIOUS BOOKMARKS



Images from the field
FLICKR PHOTOGRAPHS



The house of Spinnecollison
TWITTER HAPPIER



Spurring old acquaintances
BLOODY FACEBOOK



2

Established Nottingham 2003
THE CELEBRATED NEW MISCELLANY OF
MR. SIMON COLLISON
A.K.A. COLLY

Bested for your pleasure
POTTED AUTOBIOGRAPHY



Hello. I'm a freelance designer, speaker, and author based in Nottingham, England. I've written a few *books*, and I love doing *presentations* and workshops, plus occasional *interviews*. [Read More](#) →

Dropping science like it's hot
THE SPLENDID JOURNAL



S New Adventures. Last week, I announced New Adventures in Web Design, a unique and affordable web event that will take place in Nottingham, England. [A... More](#) →

Catalogued nocturnal matter
EXHAUSTIVE ARCHIVES



809—*Carry On Responsively*
808—*Gloucestershire Festival 2010*
807—*Memories of another trip*
806—*Thoughts on FOWD 2010*
805—*Design To Communicate*

Mr. Collison is currently
AVAILABLE FOR HIRE



Options & queries this way. Drop me a line if you wish. I'm currently considering all new projects, including design and development, writing, presentations, workshops, and consultation. →

EXTERNAL REFERENCES ([VIEW ALL](#))

Playing on the grasshopper
LAST.FM SCRIBBLES



Stashin' items from other folk
DELICIOUS BOOKMARKS



3

Established Nottingham 2003
THE CELEBRATED NEW MISCELLANY OF
MR. SIMON COLLISON
A.K.A. COLLY

Bested for your pleasure
POTTED AUTOBIOGRAPHY



Hello. I'm a freelance designer, speaker, and author based in Nottingham, England. I've written a few *books*, and I love doing *presentations* and workshops, plus occasional *interviews*. [Read More](#) →

Dropping science like it's hot
THE SPLENDID JOURNAL



S New Adventures. Last week, I announced New Adventures in Web Design, a unique and affordable web event that will take place in Nottingham, England. [A... More](#) →

Catalogued nocturnal matter
EXHAUSTIVE ARCHIVES



809—*Carry On Responsively*
808—*Gloucestershire Festival 2010*
807—*Memories of another trip*
806—*Thoughts on FOWD 2010*
805—*Design To Communicate*

Медиа запрос

В общем случае медиа-запрос состоит из ключевого слова, описывающего тип устройства (необязательный параметр) и выражения, проверяющего характеристики данного устройства. Из всех характеристик чаще всего проверяется ширина устройства **width**.

Медиа-запрос является логическим выражением, которое возвращает истину или ложь.

Добавление медиа-запросов

1. HTML

```
<link rel="stylesheet" media="screen and (color)" href="example.css">
```

2. С помощью правила @import внутри элемента `<style>` или внешней таблицы стилей

```
@import url(color.css) screen and (color);
```

3. В коде страницы:

```
<style>
@media (max-width: 600px) {
  #sidebar {display: none;}
}
</style>
```

4. Через таблицу стилей style.css

```
@media (max-width: 600px) {
  #sidebar {display: none;}
}
```

Таблица стилей, прикрепленная через тег `<link>`, будет загружаться вместе с документом, даже если её медиа-запрос вернет ложь.

Для поддержки медиа-запросов в старых браузерах можно воспользоваться JavaScript-библиотекой `css3-mediaqueries.js`, доступную по адресу <https://code.google.com/archive/p/css3-mediaqueries-js/>.

Логические операторы

С помощью логических операторов можно создавать комбинированные медиазапросы, в которых будет проверяться соответствие нескольким условиям.

Оператор and

Оператор **and** связывает друг с другом разные условия:

```
@media screen and (max-width: 600px) {  
/* CSS-стили */;  
}
```

Стили этого запроса будут применяться только для экранных устройств с шириной области просмотра не более 600px.

```
@media (min-width: 600px) and (max-width: 800px) {  
/* CSS-стили */;  
}
```

Стили этого запроса будут применяться для всех устройств при ширине области просмотра от 600px до 800px включительно.

Правило `@media all and (max-width: 600px) { ... }` равнозначно правилу `@media (max-width: 600px) { ... }`.

Оператор запятая

Оператор запятая работает по аналогии с логическим оператором `or`.

```
@media screen, projection {  
  /* CSS-стили */;  
}
```

В данном случае CSS-стили, заключенные в фигурные скобки, сработают только для экранных или проекционных устройств.

Оператор not

Оператор `not` позволяет сработать медиазапросу в противоположном случае. Ключевое слово `not` добавляется в начало медиазапроса и применяется ко всему запросу целиком, т.е. запрос

```
@media not all and (monochrome) {...}
```

будет эквивалентен запросу

```
@media not (all and (monochrome)) {...}
```

Если медиазапрос составлен с использованием оператора запятая, то отрицание будет распространяться только на ту часть, которая идет до запятой, т.е. запрос

```
@media not screen and (color), print and (color)
```

будет эквивалентен запросу

```
@media (not (screen and (color))), print and (color)
```

Оператор only

Оператор **only** используется, чтобы скрыть стили от старых браузеров (поддерживающих синтаксис медиа-запросов CSS2).

```
media="only screen and (min-width: 401px) and (max-width: 600px) "
```

Эти браузеры ожидают список медиа-типов, разделённых запятыми. И, согласно спецификации, они должны отсекают каждое значение непосредственно перед первым неалфавитно-цифровым символом, который не является дефисом. Таким образом, старый браузер должен интерпретировать предыдущий пример как `media="only"`. Поскольку данного типа медиа-типа не существует, то и таблицы стилей будут игнорироваться.

Тип носителя

Тип носителя представляет собой тип устройства, например, принтеры, экраны.

all Подходит для всех типов устройств.

print Предназначен для страничных материалов и документов, просматриваемых на экране в режиме предварительного просмотра печати.

screen Предназначен в первую очередь для экранов цветных компьютерных мониторов.

speech Предназначен для синтезаторов речи.

CSS2.1 и Media Queries 3 определяли несколько дополнительных типов, таких как `aural`, `braille`, `embossed`, `projection`, `tty`, `tv` и `handheld`, но они приняты устаревшими в Media Queries 4 и не будут использоваться.

Характеристики носителя

К характеристикам медианосителя относятся проверяемые параметры устройства. Значения, которые используются при задании характеристик, являются контрольными точками.

width Проверяет ширину области просмотра. Значения задаются в единицах длины, px, em и т.д., например, (width: 800px). Обычно для проверки используются минимальные и максимальные значения ширины. min-width применяет правило если ширина области просмотра больше значения, указанного в запросе, max-width — ширина области просмотра меньше значения, указанного в запросе.

height Проверяет высоту области просмотра. Значения задаются в единицах длины, px, em и т.д., например, (height: 500px). Обычно для проверки используются минимальные и максимальные значения высоты. min-height применяет правило если высота области просмотра больше значения, указанного в запросе, max-height — высота области просмотра которого меньше значения, указанного в запросе.

aspect-ratio Проверяет соотношение ширины к высоте области просмотра. Широкоэкранный дисплей с соотношением сторон 16:9 может быть помечен как (aspect-ratio: 16/9). min-aspect-ratio проверяет минимальное соотношение, max-aspect-ratio — максимальное соотношение ширины к высоте области просмотра.

orientation Проверяет ориентацию области просмотра. Принимает два значения: (orientation: portrait) и (orientation: landscape).

resolution Проверяет разрешение экрана (количество пикселей). Значения также могут проверять количество точек на дюйм (dpi) или количество точек на сантиметр (dpcm), например, (resolution: 300dpi). min-resolution проверяет минимальное разрешение экрана, max-resolution — максимальное.

color Проверяет количество бит на каждый из цветовых компонентов устройства вывода. Например, (min-color: 4) означает, что экран конкретного устройства должен иметь 4-битную глубину цвета. min-color проверяет минимальное количество бит, max-color — максимальное количество бит.

color-index Проверяет количество записей в таблице подстановки цветов. В качестве значения указывается положительное число, например, (color-index: 256). min-color-index проверяет минимальное количество записей, max-color-index — максимальное количество записей.

monochrome Проверяет количество битов на пиксель монохромного устройства. Значение задается целым положительным числом, например, (min-monochrome: 8). min-monochrome проверяет минимальное количество битов, max-monochrome — максимальное количество битов.

-webkit-device-pixel-ratio Задаёт количество физических пикселей устройства на каждый CSS-пиксель.

`device-width`, `device-height`, `device-aspect-ratio` являются устаревшими API, они удалены из Media Queries Level 4.

Метатег viewport

Для управления разметкой в мобильных браузерах используется метатег `viewport`.

Изначально данный тег был представлен разработчиками Apple для браузера Safari на iOS.

Мобильные браузеры отображают страницы в виртуальном окне просмотра, которое обычно шире, чем экран устройства.

С помощью метатега `viewport` можно контролировать размер окна просмотра и масштаб.

Страницы, адаптированные для просмотра на разных типах устройств, должны содержать в разделе `<head>` метатег `viewport`.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Свойство `width` определяет виртуальную ширину окна просмотра, значение `device-width` — физическую ширину устройства.

Другими словами, `width` отражает значение `document.documentElement.clientWidth`, а `device-width` — `screen.width`.

При первой загрузке страницы свойство `initial-scale` управляет начальным уровнем масштабирования, `initial-scale=1` означает, что 1 пиксель окна просмотра = 1 пиксель CSS.



Размеры экрана, на которые следует ориентироваться

При составлении медиазапросов нужно ориентироваться на так называемые **переломные (контрольные) точки дизайна**, т.е. такие значения ширины области просмотра, в которых дизайн сайта существенно меняется, например, появляется горизонтальная полоса прокрутки. Чтобы определить эти точки, нужно открыть сайт в браузере и постепенно уменьшать область просмотра.

```
/* Smartphones (вертикальная и горизонтальная ориентация) ----- */
@media only screen and (min-width : 320px) and (max-width : 480px) {
/* стили */
}
```

```
/* Smartphones (горизонтальная) ----- */
@media only screen and (min-width: 321px) {
/* стили */
}
```

```
/* Smartphones (вертикальная) ----- */
@media only screen and (max-width: 320px) {
/* стили */
}
```

```
/* iPads (вертикальная и горизонтальная) ----- */
@media only screen and (min-width: 768px) and (max-width: 1024px) {
/* стили */
}
```

```
/* iPads (горизонтальная) ----- */
@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: landscape) {
/* стили */
}
```

```
/* iPads (вертикальная) ----- */
@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: portrait) {
/* стили */
}

/* iPad 3******/
@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) {
/* стили */
}
@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) {
/* стили */
}

/* Настольные компьютеры и ноутбуки ----- */
@media only screen and (min-width: 1224px) {
/* стили */
}

/* Большие экраны ----- */
@media only screen and (min-width: 1824px) {
/* стили */
}

/* iPhone 4 ----- */
@media only screen and (min-width: 320px) and (max-width: 480px) and (orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) {
/* стили */
}
@media only screen and (min-width: 320px) and (max-width: 480px) and (orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) {
/* стили */
}

/* iPhone 5 ----- */
@media only screen and (min-width: 320px) and (max-height: 568px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 320px) and (max-height: 568px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
}
```

```
/* iPhone 6 ----- */
@media only screen and (min-width: 375px) and (max-height: 667px) and (orientation: landscape) and (-webkit-device-
pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 375px) and (max-height: 667px) and (orientation: portrait) and (-webkit-device-
pixel-ratio: 2){
/* стили */
}
/* iPhone 6+ ----- */
@media only screen and (min-width: 414px) and (max-height: 736px) and (orientation: landscape) and (-webkit-device-
pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 414px) and (max-height: 736px) and (orientation: portrait) and (-webkit-device-
pixel-ratio: 2){
/* стили */
}
/* Samsung Galaxy S3 ----- */
@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-
pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-
pixel-ratio: 2){
/* стили */
}
/* Samsung Galaxy S4 ----- */
@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-
pixel-ratio: 3){
/* стили */
}
@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-
pixel-ratio: 3){
/* стили */
}
/* Samsung Galaxy S5 ----- */
@media only screen and (min-width: 360px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-
pixel-ratio: 3){
/* стили */
}
@media only screen and (min-width: 360px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-
pixel-ratio: 3){
/* стили */
}
}
```

```
@media screen and (orientation: landscape) {  
  #logo {  
    background: url (logo1.png) no-repeat;  
  }  
}  
@media screen and (orientation: portrait) {  
  #logo {  
    background: url (logo2.png) no-repeat;  
  }  
}
```

<http://mattkersley.com/responsive/>

Стратегии использования медиа-запросов

Для создания дизайна, позволяющего лучшим образом отображать сайт на различных устройствах, используют общие стратегии медиа-запросов:

- 1) Уменьшение количества колонок (столбцов) и постепенная отмена обтекания для мобильных устройств.
- 2) Использование свойства `max-width` вместо `width` при задании ширины блока-контейнера.
- 3) Уменьшение полей и отступов на мобильных устройствах (например, нижних отступов между заголовком и текстом, левого отступа для списков и т.п.).
- 4) Уменьшение размеров шрифтов для мобильных устройств.
- 5) Превращение линейных навигационных меню в раскрывающиеся.
- 6) Скрытие второстепенного содержимого на мобильных устройствах с помощью `display: none`.
- 7) Подключение фоновых изображений уменьшенных размеров.

На мобильных устройствах не отображать фоновые изображения или видео

CSS3-фильтры

CSS3-фильтры воспроизводят в браузере визуальные эффекты, похожие на фильтры Photoshop. Фильтры можно добавлять не только к изображениям, но и к любым непустым элементам.

Набор фильтров не ограничивается предустановленным в браузере. Вы также можете использовать фильтры SVG, загрузив их вместе с элементом `svg`.

Изначально фильтры были созданы как часть спецификации SVG. Их задачей было применение эффектов, основанных на пиксельной сетке к векторной графике. С поддержкой SVG браузерами появилась возможность использовать эти эффекты непосредственно в браузерах.

Браузеры обрабатывают страницу попиксельно применяя заданные эффекты и отрисовывают результат поверх оригинала.

Таким образом, применяя несколько фильтров можно достигать различных эффектов, они как бы накладываются друг на друга. Чем больше фильтров, тем больше времени требуется браузеру, чтобы отрисовать страницу. Можно применять несколько фильтров одновременно.

Классический способ применения таких эффектов — при наведении на элемент `:hover`.

IE: не поддерживает

Edge: 13.0 кроме url()

Firefox: 35.0

Chrome: 18.0 -webkit-

Safari: 9.1, 6.0 -webkit-

Opera: 40.0, 15.0 -webkit-

iOS Safari: 9.3, 6.1 -webkit-

Android Browser: 53.0, 4.4 -webkit-

Chrome for Android: 55.0, 47.0 -webkit-

<https://codepen.io/lostsou41216364/pen/oaOWQz>

Filter

blur() Значение задается в единицах длины, например px, em. Применяет размытие по Гауссу к исходному изображению. Чем больше значение радиуса, тем больше размытие. Если значение радиуса не задано, по умолчанию берется 0. Синтаксис

filter: blur(3px);

brightness() Значение задается в % или в десятичных дробях. Изменяет яркость изображения. Чем больше значение, тем ярче изображение. Значение по умолчанию 1. Синтаксис

filter: brightness(50%);

filter: brightness(.5);

contrast() Значение задается в % или в десятичных дробях. Регулирует контрастность изображения, т.е. разницу между самыми темными и самыми светлыми участками изображения/фона. Значение по умолчанию 100%. Нулевое значение скроет исходное изображение под темно-серым фоном. Значения, увеличивающиеся от 0 до 100% или от 0 до 1, будут постепенно открывать исходное изображение до оригинального отображения, а значения свыше будут увеличивать контраст между светлыми и темными участками. Синтаксис

filter: contrast(20%);

filter: contrast(.2);

drop-shadow() Фильтр действует подобно свойствам box-shadow и text-shadow. Использует следующие значения: смещение по оси X смещение по оси Y размытость растяжение цвет тени. Отличительная особенность фильтра заключается в том, что тень добавляется к элементам и его содержимому с учетом их прозрачности, т.е. если элемент содержит текст внутри, то фильтр добавит тень одновременно для текста и видимых границ блока. В отличие от других фильтров, для этого фильтра обязательно задание параметров (минимальное - величина смещения). Синтаксис

filter: drop-shadow(2px 3px 5px black);

grayscale() Извлекает все цвета из картинки, делая на выходе черно-белое изображение. Значение задается в % или десятичных дробях. Чем больше значение, тем сильнее эффект. Синтаксис

filter: grayscale(.5); filter: grayscale(50%);

hue-rotate() Меняет цвета изображения в зависимости от заданного угла поворота в цветовом круге. Значение задается в градусах от 0deg до 360deg. 0deg - значение по умолчанию, означает отсутствие эффекта. Синтаксис

filter: hue-rotate(180deg);

invert() Фильтр делает негатив изображения. Значение задается в %. 0% не применяет фильтр, 100% полностью преобразует цвета. Синтаксис

filter: invert(100%);

opacity() Фильтр работает аналогично со свойством opacity, добавляя прозрачность элементу. Отличительная особенность - браузеры обеспечивают аппаратное ускорение для фильтра, что позволяет повысить производительность.

Дополнительный бонус - фильтр можно одновременно сочетать с другими фильтрами, создавая при этом интересные эффекты. Значение задается только в %, 0% делает элемент полностью прозрачным, а 100% не оказывает никакого эффекта. Синтаксис

filter: opacity(30%);

saturate() Управляет насыщенностью цветов, работая по принципу контрастного фильтра. Значение 0% убирает цветность, а 100% не оказывает никакого эффекта. Значения от 0% до 100% уменьшают насыщенность цвета, выше 100% - увеличивают насыщенность цвета. Значение может задаваться как в %, так и целым числом, 1 эквивалентно 100%. Синтаксис **filter: saturate(300%);**

sepia() Эффект, имитирующий старину и «ретро». Значение 0% не изменяет внешний вид элемента, а 100% полностью воспроизводит эффект сепии. Синтаксис **filter: sepia(150%);**

url() Функция принимает расположение внешнего XML-файла с svg-фильтром, или якорь к фильтру, находящемуся в текущем документе. Синтаксис **filter: url(#filterId);** /* если фильтр находится в этом документе */ **filter: url(filter.svg#filterId);** /* если фильтр с id="filterId" находится в файле filter.svg*/

none Значение по умолчанию. Означает отсутствие эффекта.

initial Устанавливает это свойство в значение по умолчанию.

inherit Наследует значение свойства от родительского элемента.

CSS3 columns

CSS3 columns — концепция многоколоночной разметки, которая позволяет разбивать контент на колонки. Колонки могут содержать заголовки, текст, таблицы, картинки и любые другие `inline-`элементы.

IE: 10.0

Firefox: 9.0 -moz-

Chrome: 4.0 -webkit-

Safari: 3.0 -webkit-

Opera: 11.1, 15.0 -webkit-

iOS Safari: 9, 7.1 -webkit-

Opera Mini: 8

Android Browser: 2.1 -webkit-

Chrome for Android: 44 -webkit-

Количество колонок `column-count`

Свойство позволяет явно задать необходимое число колонок, а их ширина будет рассчитываться, исходя из ширины доступного пространства.

По умолчанию браузер добавляет промежуток между колонками по горизонтали, равный `1em`. Если ширина изображения больше ширины колонки, то оно будет обрезано.

Если одновременно с `column-count` задается `column-width`, то значение `column-count` будет считаться максимальным числом колонок (столбцов). Свойство не наследуется.

число Предпочтительное количество колонок задаётся целым числом.

auto Значение по умолчанию, количество столбцов будет определяться другим свойством, например, `column-width`.

initial Устанавливает значение свойства в значение по умолчанию.

inherit Наследует значение свойства от родительского элемента.

```
section {  
  -webkit-column-count: 3;  
  -moz-column-count: 3;  
  column-count: 3;  
}
```

THE TRINITY FLOWER



In days of yore there was once a certain hermit, who dwelt in a cell, which he had fashioned for himself from a natural cave in the side of a hill. Now this hermit had a great love for flowers, and was moreover learned in the virtues of herbs, and in that great mystery of healing which lies hidden among the green things of God. And so it came to pass that the country people from all parts came to him for the simples which grew in the little garden which he had made before his cell. And as his fame spread, and more people came to him, he added more and more to the plot which he had reclaimed from the waste land around. But after many years there came a spring when the colors of the flowers seemed paler to the hermit than they used to be; and as summer drew on their shapes became indistinct, and he mistook one plant for another; and when autumn came, he told them by

their various scents, and by their form, rather than by sight; and when the flowers were gone, and winter had come, the hermit was quite blind. Now in the hamlet below there lived a boy who had become known to the hermit on this manner. ..

Ширина колонок column-width

Свойство позволяет разбить контент на колонки без задания свойства **column-count**. Количество колонок будет зависеть от того, сколько колонок заданной ширины может поместиться в контейнер. Не наследуется.

длина Ширина колонок задаётся в единицах длины, кроме %.

auto Значение по умолчанию, ширина колонок определяется браузером.

initial Устанавливает значение свойства в значение по умолчанию.

inherit Наследует значение свойства от родительского элемента.

```
section {  
  -webkit-column-width: 100px;  
  -moz-column-width: 100px;  
  column-width: 100px;  
}
```

Пустое пространство между колонками `column-gap`

Свойство управляет промежутком между колонками. Если для колонок установлена разделительная линия с помощью свойства `column-rule`, то эта линия будет расположена посередине промежутка, а ее ширина не изменит общую ширину. Свойство не наследуется.

длина Промежуток между колонками задается в единицах длины, кроме %.

normal Значение по умолчанию. Эквивалентно `1em`.

initial Устанавливает значение свойства в значение по умолчанию.

inherit Наследует значение свойства от родительского элемента.


```
section {  
  -webkit-column-gap: 40px;  
  -moz-column-gap: 40px;  
  column-gap: 40px;  
}
```

THE TRINITY FLOWER



In days of yore there was once a

certain hermit, who dwelt in a cell, which he had fashioned for himself from a natural cave in the side of a hill. Now this hermit had a great love for flowers, and was moreover learned in the virtues of herbs, and in that great mystery of healing which lies hidden among the green things of God. And so it came to pass that the country people from all parts came to him for the simples which grew in the little garden which he had made before his cell. And as his fame spread, and more people came to him, he added more and more to the plot which he had reclaimed from the waste land around. But after many years there came a spring when the colors of the flowers seemed paler to the hermit than they used to be; and as summer drew on their shapes became indistinct, and he mistook one plant for another; and when

autumn came, he told them by their various scents, and by their form, rather than by sight; and when the flowers were gone, and winter had come, the hermit was quite blind. Now in the hamlet below there lived a boy who had become known to the hermit on this manner. ...

Расположение элемента на несколько колонок column-span

Свойство задаёт количество колонок, которое пересечет выбранный элемент. Указывается не для блока-контейнера, а для конкретного элемента внутри, например, для заголовка.

Если нужно, чтобы изображение растянулось на все колонки, задаётся

```
img {display: block; -webkit-column-span: all; column-span: all;}. Свойство не наследуется.
```

1 Значение по умолчанию. Элемент расположен в одной колонке.

all Элемент будет распространен на все колонки.

initial Устанавливает значение свойства в значение по умолчанию.

inherit Наследует значение свойства от родительского элемента.


```
h1 {  
  -webkit-column-span: all;  
  column-span: all;  
}
```

THE TRINITY FLOWER



In days of yore there was once a certain hermit, who dwelt in a cell, which he had fashioned for himself from a natural cave in the

side of a hill. Now this hermit had a great love for flowers, and was moreover learned in the virtues of herbs, and in that great mystery of healing which lies hidden among the green things of God. And so it came to pass that the country people from all parts came to him for the simples which grew in the little garden which he had made before his cell. And as his fame spread, and more people came to him, he added more and more to the plot which he had reclaimed from the waste land around. But after many years there came a spring when the colors of the flowers seemed paler to the hermit than they used to be; and as summer drew on their shapes became indistinct, and he mistook one plant for another; and when autumn came, he told them by their various scents, and by their

form, rather than by sight; and when the flowers were gone, and winter had come, the hermit was quite blind. Now in the hamlet below there lived a boy who had become known to the hermit on this manner. ...

Разделительная линия column-rule-style

Свойство создаёт внутри пустого пространства между колонками вертикальную полосу — разделительную линию. Если не задан цвет линии, то некоторые эффекты не будут отображаться. Не наследуется.

```
section {  
  -webkit-column-rule-style: dotted;  
  -moz-column-rule-style: dotted;  
  column-rule-style: dotted;  
}
```

THE TRINITY FLOWER

In days of yore there was once a certain hermit, who dwelt in a cell, which he had fashioned for himself from a natural cave in the side of a hill. Now this hermit had a great love for flowers, and was moreover learned in the virtues of herbs, and in that great mystery of healing which lies hidden among the green things of God. And so it came to pass that the country people from all parts

came to him for the simples which grew in the little garden which he had made before his cell. And as his fame spread, and more people came to him, he added more and more to the plot which he had reclaimed from the waste land around. But after many years there came a spring when the colors of the flowers seemed paler to the hermit than they used to be; and as summer drew on their shapes became indistinct, and he

mistook one plant for another; and when autumn came, he told them by their various scents, and by their form, rather than by sight; and when the flowers were gone, and winter had come, the hermit was quite blind. Now in the hamlet below there lived a boy who had become known to the hermit on this manner. ..

`none`

Значение по умолчанию, означает отсутствие линии. Цвет и ширина, заданные для линии, игнорируются.

`hidden`

Аналогично со значением `none`, линия скрыта.

`dotted`

Отображает линию набором квадратных точек.

`dashed`

Отображает линию как последовательность из тире.

`solid`

Обычная линия.

`double`

Отображает разделительную линию в виде двух параллельных тонких линий, расположенных на некотором расстоянии между собой. Толщина разделительной линии не указывается, но сумма линий и промежутка между ними равна значению `column-rule-width`.

`groove`

Отображает линию объемной, вдавленной в полотно. Это достигается путем создания тени из двух цветов, один из которых темнее, другой — светлее.

`ridge`

Отображает разделительную линию объемной, т.е. эффект, противоположный `groove`.

`inset`

Отображает сплошную линию цветом темнее, чем заданный цвет линии.

`outset`

Отображает сплошную линию цветом, заданным свойством `column-rule-color`.

`initial`

Устанавливает значение свойства в значение по умолчанию.

`inherit`

Наследует значение свойства от родительского элемента.