

Теория

CSS-градиент представляет собой переходы от одного цвета к другому.

Градиенты создаются с помощью функций `linear-gradient()` и `radial-gradient()`.

Градиентный фон можно устанавливать в свойствах `background`, `background-image`, `border-image` и `list-style-image`.

Линейный градиент создается с помощью двух и более цветов, для которых задано направление, или **линия градиента**.

Если направление не указано, используется значение по умолчанию — **сверху-вниз**.

Цвета градиента по умолчанию распределяются равномерно в направлении, перпендикулярном линии градиента.

background: linear-gradient(угол / сторона или угол наклона с помощью ключевого слова (пары ключевых слов), первый цвет, второй цвет и т.д.);

CSS

Направление градиента может быть задано двумя способами:

с помощью угла наклона в градусах `deg`, значение которого определяет угол наклона линии внутри элемента.

```
div {  
  height: 200px;  
  background: linear-gradient(45deg, #EECFBA, #C5DDE8);  
}
```

CSS

с помощью ключевых слов `to top`, `to right`, `to bottom`, `to left`, которые соответствуют углу градиента, равному `0deg`, `90deg`, `180deg` и `270deg` соответственно.

```
div {  
  height: 200px;  
  background: linear-gradient(to right, #F6EFD2, #CEAD78);  
}
```

CSS

Если направление задано парой ключевых слов, например, `to top left`, то начальная точка градиента будет расположена в противоположном направлении, в данном случае справа внизу.

```
div {  
  height: 200px;
```

```
background: linear-gradient(to top left, powderblue, pink);
```

```
}
```

CSS

Для неравномерного распределения цветов указывается начальная позиция каждого цвета через точки остановки градиента, так называемые **color stops**. Точки остановки задаются в %, где 0% — начальная точка, 100% — конечная точка, например:

```
div {
```

```
height: 200px;
```

```
background: linear-gradient(to top, #E4AF9D 20%, #E4E4D8 50%, #A19887 80%);
```

```
}
```

CSS

Для чёткого распределения цветных полос каждый последующий цвет нужно начинать с точки остановки предыдущего цвета:

```
div {
```

```
height: 200px;
```

```
background: linear-gradient(to right, #FFDDD6 20%, #FFF9ED 20%, #FFF9ED 80%,  
#DBDBDB 80%);
```

```
}
```

1. Создать новую страницу. Установить в качестве фона градиент.

Пример css

```
background: linear-gradient(45deg, #EECFBA, #C5DDE8);
```

2. Задайте заголовок статьи. Установите правила стиля шрифта: семейство, размер. Установите заголовок по центру. Привило примените к тегу <h1>. Задайте подзаголовок статьи. Установите правила стиля шрифта: семейство, размер. Установите заголовок по центру. Привило примените к тегу <h2>.

Пример html

```
<h1 align = "center">Laba 10</h1>
```

Результат



Пример html

```
<h2 align = "center">08.10</h2>
```

Результат

08.10

3. Установите разделительные линии между заголовками `<h1>` и `<h2>`.

Результат

Laba 10

08.10

Теория

Модуль **CSS3 columns** описывает многоколоночный макет, который позволяет организовать содержимое так, чтобы оно занимало несколько вертикальных контейнеров, подобно газете или журналу. Колонки могут содержать заголовки, текст, таблицы, картинки и любые другие `inline`-элементы.

1. Количество и ширина колонок

Определение количества и ширины колонок является основополагающим при построении многоколоночного макета. Свойства `column-count` и `column-width` используются для установки количества и ширины колонок.

Третье свойство, `columns`, является сокращенным свойством, которое устанавливает ширину и количество колонок одновременно.

Другие факторы, такие как явные разрывы столбцов, содержимое и ограничения высоты, могут влиять на фактическое количество и ширину колонок.

1.1. Ширина колонок: свойство `column-width`

Свойство `column-width` указывает минимальную ширину, которую должен занимать каждый столбец.

Свойство не наследуется.

`column-width`

Значения:

`auto`

Означает, что ширина столбца будет определяться другими свойствами (например, `column-count`, если оно имеет значение, отличное от `auto`). Значение по умолчанию.

длина

Ширина колонок задаётся в единицах длины, кроме `%`.

Фактическая ширина столбца может быть больше (для заполнения доступного пространства) или уже (только если доступное пространство меньше указанной ширины столбца).

Отрицательные значения не допускаются. Используемые значения будут ограничены минимум `1px`.

`initial`

Устанавливает значение свойства в значение по умолчанию.

`inherit`

Наследует значение свойства от родительского элемента.

Синтаксис

`column-width: auto;`

`column-width: 100px;`

`column-width: 10em;`

`column-width: 3.3vw;`

`column-width: inherit;`

`column-width: initial;`

CSS

1.2. Количество колонок: свойство `column-count`

Свойство `column-count` описывает количество колонок, а их ширина будет рассчитываться, исходя из ширины доступного пространства. Если одновременно с `column-count` задается `column-width`, то значение `column-count` будет считаться максимальным числом колонок.

Свойство не наследуется.

`column-count`

Значения:

`auto`

Означает, что количество столбцов будет определяться другим свойством, например, `column-width`, если оно также не имеет значение `auto`. Значение по умолчанию.

число

Описывает максимальное количество колонок. Значение задаётся целым числом, должно быть больше `0`.

`initial`

Устанавливает значение свойства в значение по умолчанию.

`inherit`

Наследует значение свойства от родительского элемента.

Синтаксис

`column-count: auto;`

`column-count: 2;`

`column-count: inherit;`

`column-count: initial;`

CSS

1.3. Установка колонок с помощью одного свойства `columns`

Свойство `columns` — это сокращенное свойство для установки `column-width` и `column-count`. Опушенные значения устанавливаются в их начальные значения.

Свойство не наследуется.

Синтаксис

`columns: 12em; /* column-width: 12em; column-count: auto */`

`columns: auto 12em; /* column-width: 12em; column-count: auto */`

`columns: 2; /* column-width: auto; column-count: 2 */`

`columns: 2 auto; /* column-width: auto; column-count: 2 */`

`columns: auto; /* column-width: auto; column-count: auto */`

`columns: auto auto; /* column-width: auto; column-count: auto */`

`columns: inherit;`

`columns: initial;`

CSS

2. Промежутки между колонками и разделительные линии

Промежутки между колонками и разделительные линии помещаются между колонками в одном многоколоночном контейнере. Длина промежутков и разделительных равна высоте колонки. Промежутки в колонках занимают место, то есть, они раздвигают содержимое в соседних колонках.

Разделительная линии рисуется в середине промежутка между колонками, не занимая места. То есть наличие или толщина разделительной линии не изменит размещение чего-либо еще.

Линии закрашиваются чуть выше границы многоколоночного элемента. Если элемент имеет область прокрутки, разделительные линии прокручиваются вместе с колонками.

Разделительные линии отображаются только между двумя колонками, которые имеют содержимое.

2.1. Промежутки между колонками: свойство `column-gap`

Свойство `column-gap` определяет разрыв 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.

`column-gap`

Значения:

длина	Промежуток между колонками задается в единицах длины. Значения не могут быть отрицательными. Процентное значение может быть удалено из спецификации.
<code>normal</code>	Эквивалентно <code>1em</code> . Значение по умолчанию.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

Синтаксис

`column-gap: normal;`

`column-gap: 3px;`

`column-gap: 2.5em;`

`column-gap: 3%;`

`column-gap: inherit;`

`column-gap: initial;`

CSS

2.2. Цвет разделительной линии: свойство `column-rule-color`

Свойство `column-rule-color` определяет цвет разделительной линии.

Свойство не наследуется.

`column-rule-color`

Значения:

цвет	Цвет линии задается с помощью допустимых значений цвета . Значение по умолчанию <code>currentColor</code> .
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

Синтаксис

`column-rule-color: pink;`

`column-rule-color: #D71C3B;`

`column-rule-color: rgb(192, 56, 78);`

`column-rule-color: transparent;`

`column-rule-color: hsla(0, 100%, 50%, 0.6);`

`column-rule-color: inherit;`

`column-rule-color: initial;`

CSS

2.3. Стиль разделительной линии: свойство `column-rule-style`

Свойство `column-rule-style` устанавливает стиль разделительной линии.

Свойство не наследуется.

`column-rule-style`

Значения:

<code>none</code>	Значение вычисляется в <code>0</code> . Значение по умолчанию.
<code>hidden</code>	Аналогично со значением <code>none</code> , линия скрыта.
<code>dotted</code>	Отображает линию набором квадратных точек.
<code>dashed</code>	Отображает линию как последовательность из тире.

<code>solid</code>	Обычная линия.
<code>double</code>	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 <code>column-rule-width</code> .
<code>groove</code>	Отображает линию объемной, вдавленной в полотно. Это достигается путем создания тени из двух цветов, один из которых темнее, другой — светлее.
<code>ridge</code>	Отображает разделительную линию объемной, т.е. эффект, противоположный <code>groove</code> .
<code>inset</code>	Отображает сплошную линию цветом темнее, чем заданный цвет линии.
<code>outset</code>	Отображает сплошную линию цветом, заданным свойством <code>column-rule-color</code> .
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

Синтаксис

`column-rule-style: none;`

`column-rule-style: hidden;`

`column-rule-style: dotted;`

`column-rule-style: dashed;`

`column-rule-style: solid;`

`column-rule-style: double;`

`column-rule-style: groove;`

`column-rule-style: ridge;`

`column-rule-style: inset;`

`column-rule-style: outset;`

`column-rule-style: inherit;`

`column-rule-style: initial;`

CSS

2.4. Ширина разделительной линии: свойство `column-rule-width`

Свойство `column-rule-width` устанавливает ширину разделительной линии.

Отрицательные значения не допускаются. Не работает без свойства `column-rule-style`.

Свойство не наследуется.

`column-rule-width`

Значения:

<code>thin</code>	Тонкая линия.
<code>medium</code>	Значение по умолчанию. Средняя толщина линии.
<code>thick</code>	Утолщенная линия.
длина	Ширина разделительной линии задается в единицах длины.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

Синтаксис

`column-rule-width: thin;`

`column-rule-width: medium;`

`column-rule-width: thick;`

`column-rule-width: 1px;`

`column-rule-width: 2.5em;`

`column-rule-width: inherit;`

`column-rule-width: initial;`

CSS

2.5. Краткая запись свойств разделительной линии: свойство `column-rule`

Свойство `column-rule` является сокращенной записью свойств `column-rule-width` `column-rule-style` `column-rule-color`.

Свойство не наследуется.

Синтаксис

column-rule: dotted;

column-rule: solid 8px;

column-rule: solid blue;

column-rule: thick inset blue;

column-rule: inherit;

column-rule: initial;

CSS

3. Разрыв колонок

Когда содержимое размещено в нескольких колонках, браузер должен определить, где размещаются разрывы колонок. Проблема разбиения контента на колонки аналогична разбиению контента на страницы. Для решения этого вопроса было введено три новых свойства, позволяющих описывать разрывы столбцов в тех же свойствах, что и разрывы страниц: `break-before`, `break-after` и `break-inside`.

4. Используя многоколоночный макет, который позволит организовать содержимое так, чтобы оно занимало 3 вертикальных контейнера. Установить промежутки между колонками и разделительные линии (установить их ширину, цвет).

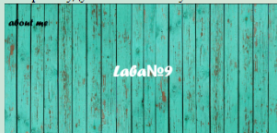
5. В колонках необходимо разместить текст статьи, иллюстрации.

Пример html

```
<div class="news">  
Задание: создайте страницу с трансформацией (см. гифку) 1. Создадим новую страницу и файл стилей. Укажем стиль для тега. Установим фоновое изображение, укажем надписи, которые будут в блоках. Результат  
1. Создадим новую страницу и файл стилей. Укажем стиль для тега. Установим фоновое изображение, укажем надписи, которые будут в блоках. Результат  
<br>  
<img src = "1.png" width = "300" height "300" />  
<br>  
2. Добавим на страницу панель с иконками, на которые установим ссылки на социальные сети. Шрифт Awesome — это коллекция масштабируемых векторных иконок. Иконки можно форматировать с помощью css-свойств, устанавливать для них цвет, размер, тень и многое другое. Шрифт версии 4.5.0 включает 605 иконок.
```

Результат

Задание: создайте страницу с трансформацией (см. гифку) 1. Создадим новую страницу и файл стилей. Укажем стиль для тега. Установим фоновое изображение, укажем надписи, которые будут в блоках. Результат



2. Добавим на страницу панель с иконками, на которые установим ссылки на социальные сети. Шрифт Awesome — это коллекция масштабируемых векторных иконок. Иконки можно форматировать с помощью css-свойств, устанавливать для них цвет, размер, тень и многое другое. Шрифт версии 4.5.0 включает 605 иконок.