MULTICOLUMN LAYOUTS

Excerpt from:
Learning Web Design, 4e

by Jennifer Robbins

Updated 2024 (browser support)

When text lines get too long, it takes a little extra effort to find the beginning of the next line. Keeping line-lengths in check is one reason magazines and newspapers divide text into columns. Web designers can divide text content into columns using the properties in the Multi-column Layout Module (*www.w3.org/TR/css3-multicol*).

The nice thing about layout columns is that, if handled correctly, they flex to fill the available space, making them responsive by default. The same bit of code can make text display in one column on narrow devices and multiple columns when there is room. On the downside, if the text is very long, it could cause readers to have to scroll back up to the top of the screen to read each subsequent column, which is not a smooth experience. Multi-column layout is best for short amounts of text or lists.

IN THIS ARTICLE

FURTHER READING

"When and How to Use CSS Multi-column Layout" by Rachel Andrew (smashingmagazine.com/2019/01/css-multiple-column-layout-multicol/)

CREATING COLUMNS

Making an element display with multiple columns is a straightforward affair, but be aware that the spec gives browsers a lot of leeway to fudge measurements to make the columns work.

There are two ways to turn an element into a multicolumn element: break it into a specific number of columns (`column-count`) or specify the width you'd like columns to be (`column-width`) and let the browser create as many

BROWSER SUPPORT NOTE

The properties used to create and span columns are well supported. Vendor prefixes are required if you need to support browsers released prior to 2017. The break-after and break-before properties have very spotty browser support. Search CanIUse.com for "multi-column layout" for updated statistics.

columns as will fit the available space. The shorthand **columns** property can be used to specify either or both of these properties.

column-count

Values: *integer*

Default: auto

Applies to: block-level elements (except table elements), table cells, and inline-block elements

Inherits: no

column-width

Values: *width*

Default: auto

Applies to: block-level elements (except table elements), table cells, and inline-block elements

Inherits: no

columns

Values: *column-width value* and/or *column-count value*

Default: see individual properties

Applies to: see individual properties

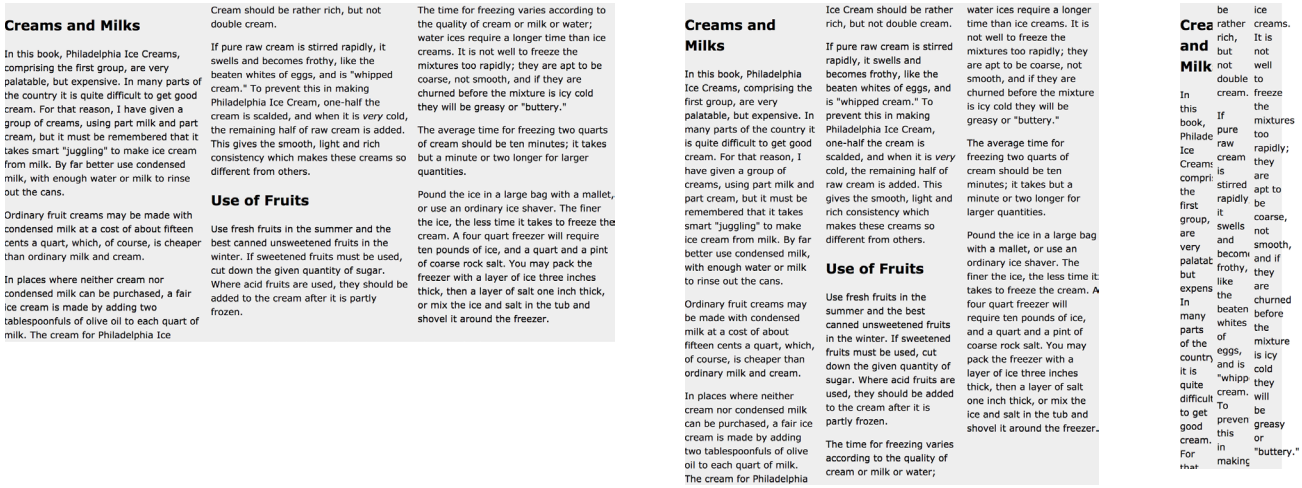
Inherits: no

Let's say we want an article always to display in three equal columns—just give the element a **column-count** of 3. The value of **column-count** must always be a positive integer, which makes sense (you can't have .02 or –3 columns). Browsers adjust the width of each column to fit the available width of the viewport, as shown in FIGURE A. I've added a gray background color to the **article** element to make its boundaries evident:

```
article {
  background-color: #eee;
  column-count: 3;
}
```

The other option is to specify the width you'd like the columns to be, perhaps based on a comfortable line length, using the **column-width** property. The browser creates as many columns as will fit in the element box (FIGURE B). If there is leftover space, the column widths expand to fill the available space, meaning the length value acts as a minimum. Notice that the narrow viewport gets just one column because that's all that fits. You can begin to see how columns are well suited for responsive layouts.

```
article {
  column-width: 250px;
}
```



Creams and Milks

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "juggling" to make ice cream from milk. By far better use condensed milk, with enough water or milk to rinse out the cans.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.

In places where neither cream nor condensed milk can be purchased, a fair ice cream is made by adding two tablespoonfuls of olive oil to each quart of milk. The cream for Philadelphia Ice Cream should be rather rich, but not double cream.

If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is *very* cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

Use of Fruits

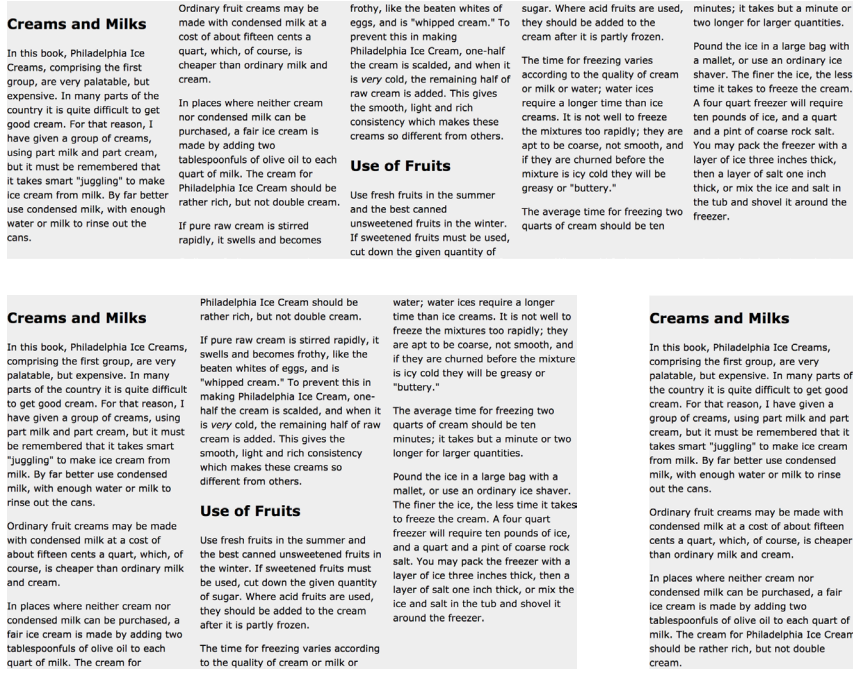
Use fresh fruits in the summer and the best canned unsweetened fruits in the winter. If sweetened fruits must be used, cut down the given quantity of sugar. Where acid fruits are used, they should be added to the cream after it is partly frozen.

The time for freezing varies according to the quality of cream or milk or water; water ices require a longer time than ice creams. It is not well to freeze the mixtures too rapidly; they are apt to be coarse, not smooth, and if they are churned before the mixture is icy cold they will be greasy or "buttery."

The average time for freezing two quarts of cream should be ten minutes; it takes but a minute or two longer for larger quantities.

Pound the ice in a large bag with a mallet, or use an ordinary ice shaver. The finer the ice, the less time it takes to freeze the cream. A four quart freezer will require ten pounds of ice, and a quart and a pint of coarse rock salt. You may pack the freezer with a layer of ice three inches thick, then a layer of salt one inch thick, or mix the ice and salt in the tub and shovel it around the freezer.

FIGURE A. The **`column-count`** property specifies the number of columns, which vary in width depending on the size of the viewport. The figure shows the same source displayed at three different viewport widths. Obviously, three columns become awkward when the viewport is narrow.



Creams and Milks

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "juggling" to make ice cream from milk. By far better use condensed milk, with enough water or milk to rinse out the cans.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.

In places where neither cream nor condensed milk can be purchased, a fair ice cream is made by adding two tablespoonfuls of olive oil to each quart of milk. The cream for Philadelphia Ice Cream should be rather rich, but not double cream.

If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is *very* cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

Use of Fruits

Use fresh fruits in the summer and the best canned unsweetened fruits in the winter. If sweetened fruits must be used, cut down the given quantity of sugar. Where acid fruits are used, they should be added to the cream after it is partly frozen.

The time for freezing varies according to the quality of cream or milk or water; water ices require a longer time than ice creams. It is not well to freeze the mixtures too rapidly; they are apt to be coarse, not smooth, and if they are churned before the mixture is icy cold they will be greasy or "buttery."

The average time for freezing two quarts of cream should be ten minutes; it takes but a minute or two longer for larger quantities.

Pound the ice in a large bag with a mallet, or use an ordinary ice shaver. The finer the ice, the less time it takes to freeze the cream. A four quart freezer will require ten pounds of ice, and a quart and a pint of coarse rock salt. You may pack the freezer with a layer of ice three inches thick, then a layer of salt one inch thick, or mix the ice and salt in the tub and shovel it around the freezer.

Creams and Milks

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "juggling" to make ice cream from milk. By far better use condensed milk, with enough water or milk to rinse out the cans.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.

In places where neither cream nor condensed milk can be purchased, a fair ice cream is made by adding two tablespoonfuls of olive oil to each quart of milk. The cream for Philadelphia Ice Cream should be rather rich, but not double cream.

FIGURE B. The **`column-width`** property specifies a minimum width for each column, and the browser calculates how many columns fit in the available space. Column widths expand to fill the full width of the element. Wider viewports allow more columns.

The Shorthand columns Property

If you'd like to save some typing and shave a few bytes off your document size, feel free to use the shorthand **columns** attribute for either or both column types. Provide an integer, and **columns** creates a number of columns:

```
columns: 3;       /* same as column-count: 3 */
```

Provide a length, and it will be used as the column width:

```
columns: 250px;  /* same as column-width: 250px; */
```

The **columns** shorthand is most useful when you're specifying both count and width. The count value is used as a maximum, which keeps the column count under control on wide screens, and the minimum width value ensures that narrow screens don't display the hot mess we saw in FIGURE A. Compare the examples using both values in FIGURE C with the width-only examples in FIGURE B.

```
columns: 3 250px;
```

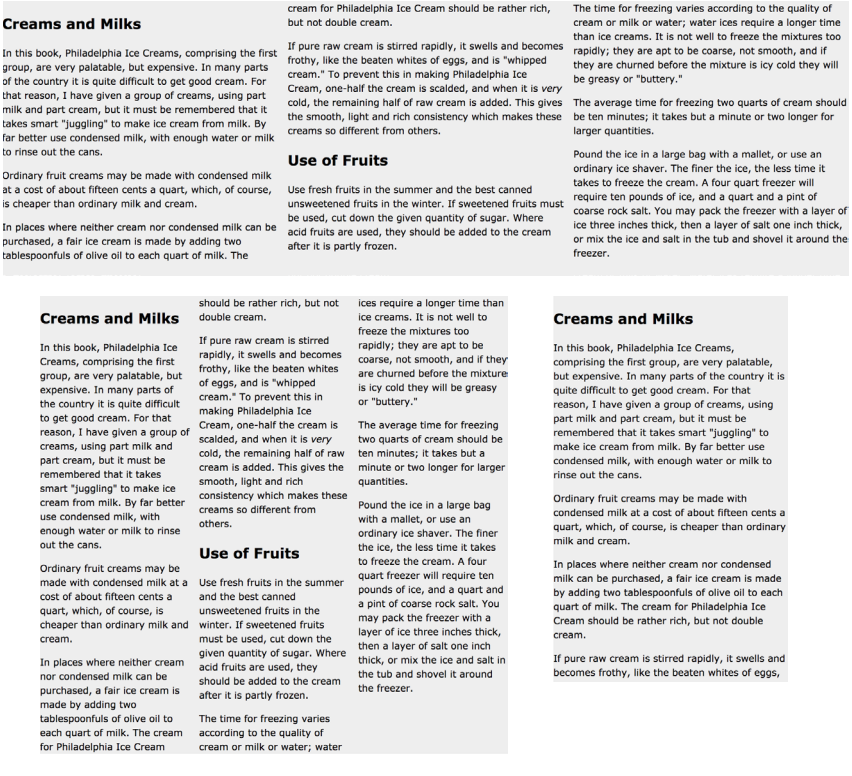


FIGURE C. The **columns** shorthand property allows you specify a count, a width, or both.

Column Mechanics

You should be getting a feel for multicolumn elements by now, but let's talk about what's happening behind the scenes. The browser divides the multicolumn element into a number of column boxes, a new type of container that exists between the element box and the actual content.

Column boxes are laid out left to right in left-to-right reading languages (and vice versa). All column boxes within the element have the same width and height (see **Note**). The height of the multicolumn element is adjusted to accommodate all of the content with text balanced across columns by default.

Column boxes behave like block elements with a few exceptions. First, you cannot apply backgrounds to individual column boxes, nor do they have any padding, border, or margin. Padding and margins applied to a multicolumn element are applied to the element box itself, not its columns. When you float an image or other element, it floats to the edge of its respective column box.

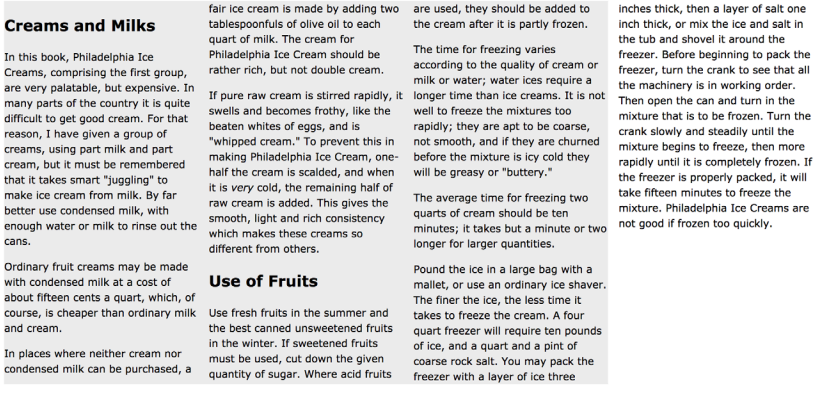
NOTE

As of this writing, all columns will be the same width. But there is the possibility that down the road we will be able to specify columns of varying width.

Handling Overflow

By default, the height of the multicolumn element expands to accommodate all of the content; however, if you restrict the height of the element (using `height` or `max-height`), the content may not fit. If that is the case, browsers create additional overflow columns outside the element box until all the content is displayed (FIGURE D). This could require users to scroll horizontally to view the newly created columns, which is considered a user experience no-no.

The easiest way to avoid surprise horizontal scrollbars is not to restrict the height of the element. You could also use a media query that sets the height of the container only when the viewport is wide enough to accommodate all the content.



Creams and Milks

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "juggling" to make ice cream from milk. By far better use condensed milk, with enough water or milk to rinse out the cans.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.

In places where neither cream nor condensed milk can be purchased, a fair ice cream is made by adding two tablespoonfuls of olive oil to each quart of milk. The cream for Philadelphia Ice Cream should be rather rich, but not double cream.

If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is *very* cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

Use of Fruits

Use fresh fruits in the summer and the best canned unsweetened fruits in the winter. If sweetened fruits must be used, cut down the given quantity of sugar. Where acid fruits are used, they should be added to the cream after it is partly frozen.

The time for freezing varies according to the quality of cream or milk or water; water ices require a longer time than ice creams. It is not well to freeze the mixtures too rapidly; they are apt to be coarse, not smooth, and if they are churned before the mixture is icy cold they will be greasy or "buttery."

The average time for freezing two quarts of cream should be ten minutes; it takes but a minute or two longer for larger quantities.

Pound the ice in a large bag with a mallet, or use an ordinary ice shaver. The finer the ice, the less time it takes to freeze the cream. A four quart freezer will require ten pounds of ice, and a quart and a pint of coarse rock salt. You may pack the freezer with a layer of ice three inches thick, then a layer of salt one inch thick, or mix the ice and salt in the tub and shovel it around the freezer. Before beginning to pack the freezer, turn the crank to see that all the machinery is in working order. Then open the can and turn in the mixture that is to be frozen. Turn the crank slowly and steadily until the mixture begins to freeze, then more rapidly until it is completely frozen. If the freezer is properly packed, it will take fifteen minutes to freeze the mixture. Philadelphia Ice Creams are not good if frozen too quickly.

FIGURE D. The browser creates extra columns outside the element box (indicated by its light-gray background color) when there is not enough room for the content.

Another case of ill-fitting content may occur when a long word or a replaced element, such as an image, is too wide to fit in its column box. When content is too wide, browsers clip the word or image at the midpoint between columns (FIGURE E).

By default, long words and images that do not fit in the column get clipped.

Clipping fixed with word break (`wbr`) element in long words and `width: 100%` for images.

FIGURE E. When elements are too wide for the column, they get clipped on the sides (left). Setting the width of images to 100% ensures they always fit within the column (right).

Set width: 100% on images so they resize to fit the width of the column automatically.

For extremely long words, inserting `wbr` elements causes the word to break at that point when the column gets too narrow (on every browser except Internet Explorer, that is):

```
<p>Supercali<wbr>fragilistic<wbr>expialidocious. …
```

For images and other non-replaced elements, setting `width: 100%` guarantees they will fill the column width exactly (FIGURE E).

SPACES BETWEEN COLUMNS

Browsers add an amount of space (typically 1 em) between columns by default, but you can control that space with the `gap` property. The `column-rule` property adds a line between columns.

gap

Values: *length* | `normal`

Default: `normal`

Applies to: multicolumn elements

Inherits: no

column-rule

Values: *column-rule-style column-rule-color column-rule-width*

Default: per individual properties

Applies to: multicolumn elements

Inherits: no

Note: `column-rule` is a shorthand property representing `column-rule-style`, `column-rule-color`, and `column-rule-width` individual properties. The `column-rule` properties take the same values as the corresponding border styles.

The `gap` property takes a length value that specifies the width of the space between columns. Unlike some *other* column properties we know, the gap always renders at the specified width. The `normal` keyword uses the browser default (usually 1 em). Let's see what happens when I increase the `column-gap` to 4 em in FIGURE F:

```
article {
  columns: 3 250px;
  gap: 4em;
}
```

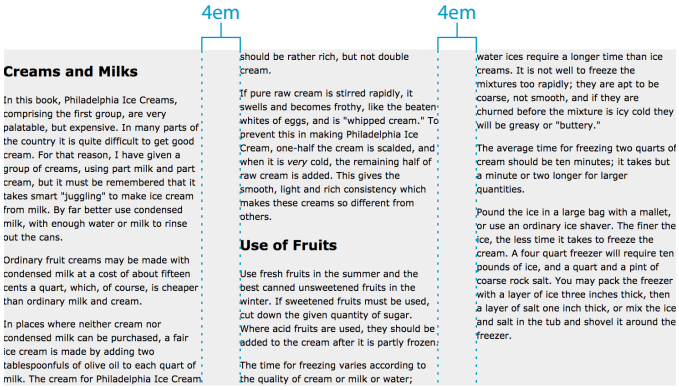


FIGURE F. Changing the space between columns with `column-gap`.

Now I'm going to add a rule between the columns as well with the `column-rule` property (FIGURE G). Column rules are positioned in the center of the column gap length. The `column-rule` property is a shorthand for three individual properties: `column-rule-style`, `column-rule-color`, and `column-rule-width`. All of these properties take the same values as their respective `border-*` properties, so if you know how to specify a border in CSS, you can do the same for column rules:

```
article {
  background-color: #eee;
  columns: 3 250px;
  gap: 4em;
  column-rule: 2px solid green;
}
```

The Too-Tall Column Problem

Another potential multicolumn pitfall is that the container may be so tall that users need to repeatedly scroll up to read the top of the next column. That's not a fun way to read. Here are suggested workarounds:

- Avoid putting extremely long text into one multicolumn element.
- Use column spans (introduced later in this chapter) to break a long text piece into smaller columned sections.
- Use a media query that checks the height of the viewport and breaks the content into columns only when there is enough room to display the text without scrolling. Of course, this works only if you know the final amount of content, and may not be a solution if the length of an article is unknown, as is the case with content management systems.

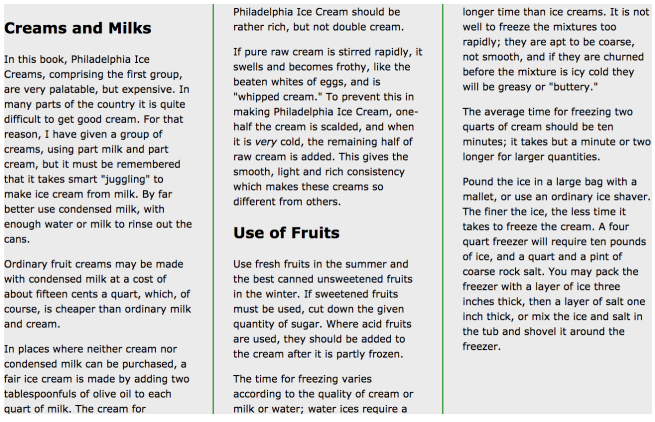


FIGURE G. Adding rules between columns with `column-rule`.

BALANCING COLUMNS

By default, browsers do their best to keep the content in columns balanced by resizing the height of the element so it fits just right. If you apply a specific height to the element, however, the content might overflow (as we saw earlier), or it might come up short. If it's short, there are two options for filling in content: keep the content balanced across columns, or fill each column sequentially until the content runs out. The `column-fill` property specifies your preference.

column-fill

Values: `auto` | `balance` | `balance-all`

Default: `balance`

Applies to: multicolumn elements

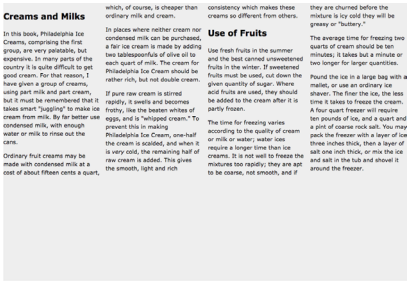
Inherits: no

The `balance` keyword tells the browser to balance the content between columns, which is the behavior browsers follow by default whether a height is specified or not. For cases in which you've set a height on the multicolumn element and want the columns in it to be filled sequentially, use the `column-fill` property with the `auto` keyword value. FIGURE H shows the difference.

NOTE

The `balance-all` *keyword is relevant to paged media (such as print documents) and balances all columns across all pages. In paged media, the* `balance` *keyword balances the columns on the last page only.*

column-fill: balance (default)



column-fill: auto

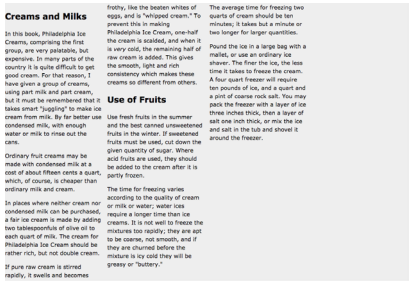


FIGURE H. The `column-fill` property set to `balance` (left) and to fill sequentially with `auto` (right).

SPANNING COLUMNS

The **column-span** property allows an element within a column to break out of the column box and span across all the columns in the multicolumn element. Spanning can be a useful tool for drawing attention to a particular element or for breaking a long, columned element into manageable chunks.

column-span

Values: all | none

Default: none

Applies to: elements within multicolumn elements

Inherits: no

Let's start with an example. In FIGURE 1, I've turned a short paragraph into a callout by giving it the class **callout**, then the **column-span: all** rule. The following HTML source shows where it appears in relation to other content in the article. I've also added some borders and a background color to make the boundaries of its element box clear.

THE MARKUP

```
<p>Ordinary fruit creams may be made with condensed milk at a cost
of about fifteen cents a quart, which, of course, is cheaper than
ordinary milk and cream.</p>

<p class="callout">The cream for Philadelphia Ice Cream should be
rather rich, but not double cream. </p>

<p>If pure raw cream is stirred rapidly, it swells and becomes frothy,
like the beaten whites of eggs, and is "whipped cream." To prevent
this in …
```

THE STYLES

```
p.callout {
  column-span: all;
  font-weight: bold;
  font-size: 1.2em;
  background-color: #c6de89; /* light green */
  border-top: 1px solid black;
  border-bottom: 1px solid black;
  padding: .5em;
}
```

The arrows on the figure show how the content flows into the columns. The spanning element essentially stops the column flow before it, then restarts the columned layout again with the following element.

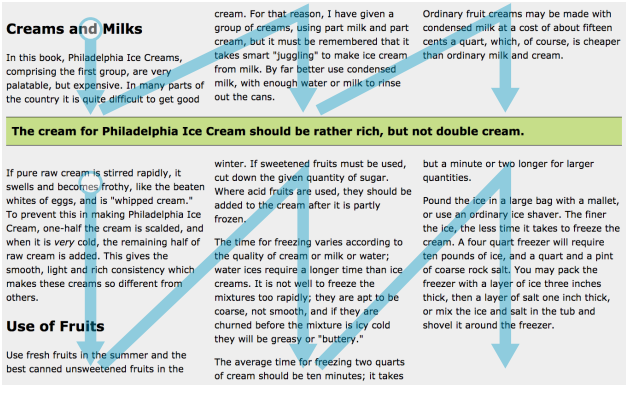


FIGURE 1. The **column-span** property makes the selected paragraph span across all of the columns to be used as a callout or pull quote. The column layout starts again after the span.

BREAKING CONTENT WITHIN COLUMNS

When content gets poured into columns automatically, there is the potential for some pretty awkward breaking points in the text. Some designers may be irked by a subhead at the bottom of the column with its respective text starting at the top of the next column. Or they may not want column breaks to occur inside a long heading or a list.

CSS3 introduced three properties that let authors control where content should break: **break-before**, **break-after**, and **break-inside** (see **Note**). Unfortunately, they are not well supported as of this writing, with support lacking in Safari (macOS and iOS) and Firefox. Check CanIUse.com for updated support information.

NOTE

The generic **break-*** *properties replace the more specific* **page-break-before**, **page-break-after**, *and* **page-break-inside** *properties from CSS 2.1. The advantage is that they can be used for columns and regions in addition to pages.*

break-after

Values: auto | always | left | right | recto | verso | page | column | region | avoid | avoid-page | avoid-column | avoid-region

Default: auto

Applies to: block-level elements

Inherits: no

break-before

Values: auto | always | left | right | recto | verso | page | column | region | avoid | avoid-page | avoid-column | avoid-region

Default: auto

Applies to: block-level elements

Inherits: no

break-inside

Values: auto | avoid | avoid-page | avoid-column | avoid-region

Default: auto

Applies to: block-level elements

Inherits: no

As you can see, there are a whole lot of values for the **break-*** properties, most of which apply only to paged media. The values we are interested in here are **column** and **avoid-column**.

The **break-before** property inserts a column break before the selected element. One application of **break-before** is to make sure that subheads always start on new columns and stay with their respective text. In this example,

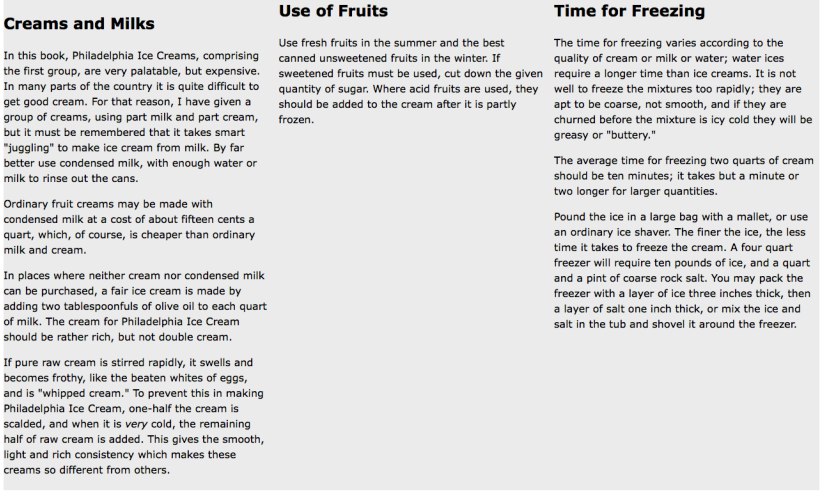
I've set columns to always break before **h2** headings. The result is shown in FIGURE J.

```
h2 {
  break-before: column;
}
```

To force a column break after a particular element, simply use **break-after: column**. The CSS spec provides the **avoid-column** value for ensuring that column breaks *don't* happen before or after particular elements, but there is very little browser support for it.

You may also decide that you don't want a column break to happen *inside* an element, such as a long headline, a list, or a quote. To prevent column breaks inside an element, apply **break-inside: avoid-column**.

That wraps up our tour of multicolumn layout properties. You've learned how to specify the number and width of columns, adjust the space or add a rule between them, specify how columns are filled, and allow certain elements to span across the columns.



Creams and Milks

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "juggling" to make ice cream from milk. By far better use condensed milk, with enough water or milk to rinse out the cans.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.

In places where neither cream nor condensed milk can be purchased, a fair ice cream is made by adding two tablespoonfuls of olive oil to each quart of milk. The cream for Philadelphia Ice Cream should be rather rich, but not double cream.

If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is *very* cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

Use of Fruits

Use fresh fruits in the summer and the best canned unsweetened fruits in the winter. If sweetened fruits must be used, cut down the given quantity of sugar. Where acid fruits are used, they should be added to the cream after it is partly frozen.

Time for Freezing

The time for freezing varies according to the quality of cream or milk or water; water ices require a longer time than ice creams. It is not well to freeze the mixtures too rapidly; they are apt to be coarse, not smooth, and if they are churned before the mixture is icy cold they will be greasy or "buttery."

The average time for freezing two quarts of cream should be ten minutes; it takes but a minute or two longer for larger quantities.

Pound the ice in a large bag with a mallet, or use an ordinary ice shaver. The finer the ice, the less time it takes to freeze the cream. A four quart freezer will require ten pounds of ice, and a quart and a pint of coarse rock salt. You may pack the freezer with a layer of ice three inches thick, then a layer of salt one inch thick, or mix the ice and salt in the tub and shovel it around the freezer.

FIGURE J. When the **break-before** property is applied to **h2**s, the columns are broken before the headings.

WARNING

If you have more subheads than columns, the browser creates overflow columns outside the multicolumn element.

CSS REVIEW: MULTICOLUMN PROPERTIES

Property	Description
`break-after`	Indicates whether a column should break after the element
`break-before`	Indicates whether a column should break before the element
`break-inside`	Indicates whether a column should break inside the element
`column-count`	Specifies the number of columns an element should be divided into
`column-fill`	Shorthand property for rounding the corners of the visible element box
`column-gap`	Specifies the width of the gap between columns

`column-rule`
`column-rule-style`
`column-rule-width`
`column-rule-color`
	The style of the rule centered between columns; **`column-rule`** is a shorthand for the other three

`column-span`	Specifies how many columns an element should span across
`column-width`	Suggested, optimal width for a column
`columns`	Specifies the border width for each side of the element