
Technologies ▼

References & Guides ▼

Feedback ▼

Sign in 

 Search

Multiple-column layout

Previous

Overview: CSS layout

Next

The multiple-column layout specification gives you a method of laying content out in columns, as you might see in a newspaper. This article explains how to use this feature.

Prerequisites: HTML basics (study [Introduction to HTML](#)), and an idea of How CSS works (study [Introduction to CSS](#).)

Objective: To learn how to create multiple-column layout on webpages, such as you might find in a newspaper.

A basic example

We will now explore how to use multiple-column layout, often referred to as *multicol*. You can follow along by downloading the multicol starting point file and adding the CSS into the appropriate places. At the bottom of the section you can see a live example of what the final code should look like.

Our starting point contains some very simple HTML; a wrapper with a class of container inside which is a heading and some paragraphs.

The `<div>` with a class of container will become our multicol container. We switch on multicol by using one of two properties `column-count` or `column-width`. The `column-count` property will create as many columns as the value you give it, so if you add the following CSS to your stylesheet and reload the page, you will get three columns:

```
1 .container {  
2   column-count: 3;  
3 }
```

The columns that you create have flexible widths — the browser works out how much space to assign each column.

```
1 <div class="container">  
2   <h1>Simple multicol example</h1>  
3  
4   <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus a  
5   Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor i  
6   Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae conval  
7   ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis  
8   quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum  
9  
10  <p>Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuad  
11  elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lob  
12  cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdu  
13  dis parturient montes, nascetur ridiculus mus.</p>  
14 </div>
```

```
1 .container {  
2   column-count: 3;  
3 }
```

Simple multicol example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras norttitor

Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum

est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget

Change your CSS to use `column-width` as follows:

```
1 | .container {  
2 |     column-width: 200px;  
3 | }
```

The browser will now give you as many columns as it can of the size that you specify; any remaining space is then shared between the existing columns. This means that you will not get exactly the width that you specify, unless your container is exactly divisible by that width.

```
1 | .container {  
2 |     column-width: 200px;  
3 | }
```

Simple multicol example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam

Styling the columns

The columns created by multicol cannot be styled individually. There is no way to make one column bigger than other columns, or to change the background or text color of a single column. You have two opportunities to change the way that columns display:

- Changing the size of the gap between columns using the `column-gap`.
- Adding a rule between columns with `column-rule`.

Using your example above, change the size of the gap by adding a `column-gap` property:

```
1 | .container {  
2 |     column-width: 200px;  
3 |     column-gap: 20px;  
4 | }
```

You can play around with different values — the property accepts any length unit. Now add a rule between the columns, with `column-rule`. In a similar way to the `border` property that you encountered in previous lessons, `column-rule` is a shorthand for `column-rule-color`, `column-rule-style`, and `column-rule-width`, and accepts the same values as `border`.

```
1 .container {  
2   column-count: 3;  
3   column-gap: 20px;  
4   column-rule: 4px dotted rgb(79, 185, 227);  
5 }
```

Try adding rules of different styles and colors.

Simple multicol example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras

ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id

Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam

Something to take note of is that the rule does not take up any width of its own. It lies across the gap you created with `column-gap`. To make more space either side of the rule you will need to increase the `column-gap` size.

Columns and fragmentation

The content of a multi-column layout is fragmented. It essentially behaves the same way as content behaves in paged media — such as when you print a webpage. When you turn your content into a multicol container it is fragmented into columns, and the content breaks to allow this to happen.

Sometimes, this breaking will happen in places that lead to a poor reading experience. In the live example below, I have used multicol to lay out a series of boxes, each of which have a heading and some text inside. The heading becomes separated from the text if the columns fragment between the two.

```
1  <div class="container">
2    <div class="card">
3      <h2>I am the heading</h2>
4      <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus
5          vulputate. Duis felis orci, pulvinar id metus ut, rutrum luct
6          tellus laoreet sit amet. Sed auctor cursus massa at porta. In
7          vel, viverra egestas ligula.</p>
8    </div>
9
10   <div class="card">
11     <h2>I am the heading</h2>
12     <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luct
13         vulputate. Duis felis orci, pulvinar id metus ut, rutrum luct
14         tellus laoreet sit amet. Sed auctor cursus massa at porta. In
15         vel, viverra egestas ligula.</p>
16   </div>
17
18   <div class="card">
19     <h2>I am the heading</h2>
20     <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luct
21         vulputate. Duis felis orci, pulvinar id metus ut, rutrum luct
22         tellus laoreet sit amet. Sed auctor cursus massa at porta. In
23         vel, viverra egestas ligula.</p>
24   </div>
25   <div class="card">
26     <h2>I am the heading</h2>
27     <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luct
```

```
28         vulputate. Duis felis orci, pulvinar id metus ut, rutrum luct
29         tellus laoreet sit amet. Sed auctor cursus massa at porta. In
30         vel, viverra egestas ligula.</p>
31     </div>
32
33     <div class="card">
34         <h2>I am the heading</h2>
35         <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luct
36         vulputate. Duis felis orci, pulvinar id metus ut, rutrum luct
37         tellus laoreet sit amet. Sed auctor cursus massa at porta. In
38         vel, viverra egestas ligula.</p>
39     </div>
40
41     <div class="card">
42         <h2>I am the heading</h2>
43         <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luct
44         vulputate. Duis felis orci, pulvinar id metus ut, rutrum luct
45         tellus laoreet sit amet. Sed auctor cursus massa at porta. In
46         vel, viverra egestas ligula.</p>
47     </div>
48
49     <div class="card">
50         <h2>I am the heading</h2>
51         <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luct
52         vulputate. Duis felis orci, pulvinar id metus ut, rutrum luct
53         tellus laoreet sit amet. Sed auctor cursus massa at porta. In
54         vel, viverra egestas ligula.</p>
55     </div>
56
57 </div>
```

```
1  .container {
2      column-width: 250px;
3      column-gap: 20px;
4  }
5
6  .card {
7      background-color: rgb(207, 232, 220);
8      border: 2px solid rgb(79, 185, 227);
```

```
9 padding: 10px;
10 margin: 0 0 1em 0;
11 }
```

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci.

To control this behavior we can use properties from the CSS Fragmentation specification. This specification gives us properties to control breaking of content in multicol and in paged media. For example, add the property `break-inside` with a value of `avoid` to the rules for `.card`. This is the container of the heading and text, and therefore we do not want to fragment this box.

At the present time it is also worth adding the older property `page-break-inside: avoid` for best browser support.


```
1  .card {
2    break-inside: avoid;
3    page-break-inside: avoid;
4    background-color: rgb(207,232,220);
5    border: 2px solid rgb(79,185,227);
6    padding: 10px;
7    margin: 0 0 1em 0;
8  }
```

Reload the page and your boxes should stay in one piece.

```
1  .container {
2    column-width: 250px;
3    column-gap: 20px;
4  }
5
6  .card {
7    break-inside: avoid;
8    page-break-inside: avoid;
9    background-color: rgb(207, 232, 220);
10   border: 2px solid rgb(79, 185, 227);
11   padding: 10px;
12   margin: 0 0 1em 0;
13 }
```

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

I am the heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula.

Summary

You now know how to use the basic features of multiple-column layout, another tool at your disposal when choosing a layout method for the designs you are building.

See also

- [CSS Fragmentation](#)
- [Using multi-column layouts](#)

[Previous](#)[Overview: CSS layout](#)[Next](#)

In this module

- [Introduction to CSS layout](#)
 - [Normal Flow](#)
 - [Flexbox](#)
 - [Grid](#)
 - [Floats](#)
 - [Positioning](#)
 - [Multiple-column Layout](#)
 - [Legacy Layout Methods](#)
 - [Supporting older browsers](#)
 - [Fundamental Layout Comprehension Assessment](#)
-