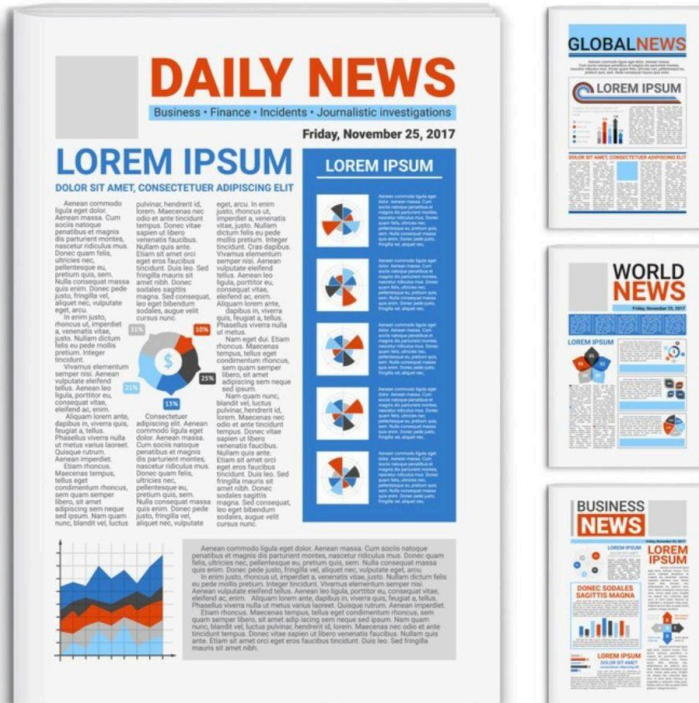


CSS Multi-Column Layout



Abdelfattah Ragab

CSS Multi-Column Layout

Abdelfattah Ragab

Introduction

Welcome to the book "CSS Multi-Column Layout". In this book I will explain how you can use the Multi-Column Layout module. It is the third layout module after the Grid and Flex layout. The multi-column layout is specifically used for newspapers and magazines to create elegant and responsive columns where the content automatically fills the columns. You will learn how to create the columns and how to set the properties of rules, span, column breaks, etc... By the end of this book, you will have full control over the multi-column layout and be able to master all possible scenarios. Let's go!

key concepts

CSS3 Multi-column Layout is a feature that allows you to create a text layout with multiple columns, similar to a newspaper or a magazine. It provides a way to divide content into multiple columns, making it easier to read and navigate through lengthy text. Here's an overview of CSS3 Multi-column Layout and its key concepts:

Creating Multi-column Layout

- To create a multi-column layout, you need to apply the column-count property to the container element.
- Specify the number of columns you want by setting the value of column-count to an integer.
- Alternatively, you can use the column-width property to set the desired width of each column. The number of columns will be determined automatically based on the available space.

Column Gap

- The column-gap property is used to define the space between columns.
- You can set the value to a length (e.g., 10px) or a percentage (e.g., 5%).
- By default, the browser determines the appropriate gap size.

Column Rules

- You can add a rule (line) between columns using the column-rule property.
- The column-rule property sets the width, style, and color of the rule.
- This can help visually separate columns.

Column Span

- The column-span property allows an element to span across multiple columns.
- By setting column-span: all;, an element can span the full width of the multi-column layout.

- However, note that not all browsers support column spanning.

Column Fill

- The column-fill property controls how content is distributed across columns.
- By default, content is balanced evenly across columns (column-fill: balance;).
- Alternatively, you can use column-fill: auto; to let content flow naturally and potentially create unequal column heights.

Column Breaks

- CSS3 Multi-column Layout automatically handles column breaks based on available space.
- You can control how column breaks occur using the break-before, break-after, and break-inside properties.
- These properties allow you to specify whether column breaks are allowed before or after an element or within an element.

CSS3 Multi-column Layout provides an effective way to organize text content into multiple columns, improving readability and user experience. It's particularly useful for presenting articles, blog posts, and other text-heavy content. By applying the appropriate properties, you can customize the column count, gap, rules, and spanning to create visually appealing and well-structured multi-column layouts.

The CSS multi-column layout module lets you divide content across multiple columns.

column-count

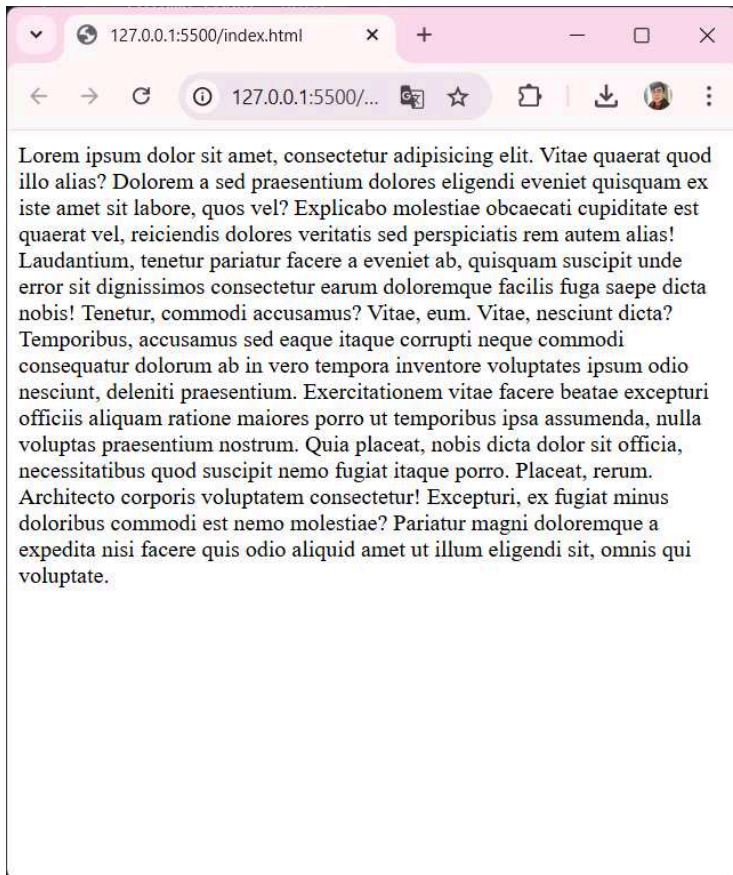
The column-count property specifies the number of columns an element should be divided into.

Values

- auto
- number

auto

Default value. The number of columns is determined by other properties, such as column-width



```
<style>
  .wrapper {
    column-count: auto;
  }
</style>
<div class="wrapper">
  <p>
```


Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Vitae quaerat quod

 illo alias? Dolorem a sed praesentium
dolores eligendi eveniet quisquam ex

 iste amet sit labore, quos vel? Explicabo
molestiae obcaecati cupiditate est

 quaerat vel, reiciendis dolores veritatis
sed perspiciatis rem autem alias!

 Laudantium, tenetur pariatum facere a
eveniet ab, quisquam suscipit unde

 error sit dignissimos consectetur earum
doloremque facilis fuga saepe dicta

 nobis! Tenetur, commodi accusamus? Vitae,
eum. Vitae, nesciunt dicta?

 Temporibus, accusamus sed eaque itaque
corrupti neque commodi consequatur

 dolorum ab in vero tempora inventore
voluptates ipsum odio nesciunt,

 deleniti praesentium. Exercitationem vitae
facere beatae excepturi officiis

 aliquam ratione maiores porro ut temporibus
ipsa assumenda, nulla voluptas

 praesentium nostrum. Quia placeat, nobis
dicta dolor sit officia,

 necessitatibus quod suscipit nemo fugiat
itaque porro. Placeat, rerum.

Architecto corporis voluptatem consecetur!
Excepturi, ex fugiat minus
doloribus commodi est nemo molestiae?
Pariatur magni doloremque a expedita
nisi facere quis odio aliquid amet ut illum
eligendi sit, omnis qui
voluptate.

</p>

</div>

number

The optimum number of columns into which the content
of the element should flow