

CSS Multi-Column Layouts

Purpose:

Used to divide text or content into **multiple columns** (like a newspaper or magazine layout).

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Multi-Column Layout</title>
  <style>
    .news {
      column-count: 3;      /* number of columns */
      column-gap: 30px;     /* space between columns */
      column-rule: 2px solid gray; /* vertical line between columns */
      font-size: 18px;
      text-align: justify;
    }
  </style>
</head>
<body>
  <div class="news">
    The multi-column layout module makes it easy to define multiple columns of text just like
    in newspapers.
    You can control the number of columns, the space between them, and even add a rule
    (line) between columns.
    This technique is best for content-heavy pages such as blogs or magazines where reading
    flow is important.
  </div>
</body>
</html>
```

Explanation:

- `column-count` sets how many columns.
- `column-gap` sets space between columns.
- `column-rule` adds a dividing line between them.
- Perfect for **articles, blogs, and news content**.

2. Float and Clear (Legacy Layout Technique)

Purpose:

Before Flexbox and Grid, web designers used **float** to position elements side by side (like image with text).

clear is used to **prevent overlap** or unwanted wrapping.

xample:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Float and Clear Example</title>
  <style>
    .container {
      width: 600px;
      border: 2px solid #333;
      padding: 10px;
    }

    img {
      float: left; /* pushes image to left side */
      width: 150px;
      margin: 10px;
    }

    p {
      text-align: justify;
    }

    .footer {
      clear: both; /* stops floating from affecting footer */
      background: lightgray;
      text-align: center;
      padding: 10px;
      margin-top: 10px;
    }
  </style>
</head>
<body>
  <div class="container">
    
    <p>
      The float property allows text and inline elements to wrap around an element, like this image.
    <p>
      This technique was once used for creating entire page layouts before Flexbox and Grid.
      The clear property ensures that following elements (like footers) don't overlap floating items.
    </p>
  </div>
</body>
```

```
<div class="footer">Footer (cleared below the image)</div>
</div>
</body>
</html>
```

Explanation:

- float: left; or float: right; moves elements to one side.
 - clear: both; prevents overlap and moves the element below floated items.
 - **Outdated** for modern layouts — use **Flexbox or Grid** instead.
-

3. object-fit and object-position

Purpose:

These properties control how **images or videos** fit inside a container, similar to how background-size works for background images.

They are especially useful for responsive images in cards, galleries, or hero sections.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>object-fit and object-position</title>
<style>
.gallery {
  display: flex;
  gap: 20px;
  justify-content: center;
  margin-top: 30px;
}

.img-box {
  width: 200px;
  height: 150px;
  border: 3px solid #555;
  overflow: hidden;
}

.cover img {
  object-fit: cover; /* fills box, crops excess */
  width: 100%;
  height: 100%;
```

```

}

.contain img {
  object-fit: contain; /* fits entire image inside box */
  width: 100%;
  height: 100%;
}

.position img {
  object-fit: cover;
  object-position: top; /* focus on top part of image */
  width: 100%;
  height: 100%;
}
</style>
</head>
<body>
  <div class="gallery">
    <div class="img-box cover"></div>
    <div class="img-box contain"></div>
    <div class="img-box position"></div>
  </div>
</body>
</html>

```

Explanation:

Property	Description	Result
object-fit: cover;	Fills the container and crops overflow	Good for hero images
object-fit: contain;	Shows the full image with padding	Good for logos
object-position: top;	Focuses image on top area	Good for portraits

Feature	Purpose	Modern Use	Example Property
Multi-column	Split text into multiple columns	Magazines, articles	column-count, column-gap
Float & Clear	Legacy side-by-side layout	Simple text wrapping	float, clear
Object-fit & Object-position	Control how media fits container	Responsive images, galleries	object-fit: cover, object-position: top

