


```
.clearfix::after {
    content: "";
    clear: both;
    display: table;
}
</style>
</head>
<body>
<h2>Grid of Boxes</h2>
<p>Float boxes side by side:</p>
<div class="clearfix">
  <div class="box" style="background-color:#bbb">
    <p>Some text inside the box.</p>
  </div>
  <div class="box" style="background-color:#ccc">
    <p>Some text inside the box.</p>
  </div>
  <div class="box" style="background-color:#ddd">
    <p>Some text inside the box.</p>
  </div>
</div>
<p>
  Note that we also use the clearfix hack to take care of the layout
  flow, and that we add the box-sizing property to make sure that the box doesn't
  break due to extra padding. Try to remove this
  code to see the effect.
</p>
</body>
</html>
```

3
(,) 가 , . .
"box-sizing" (가) , .

Images Side By Side

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Images Side By Side</title>
  <style>
    * {
      box-sizing: border-box;
    }

    .img-container {
      width: 33.33%;
      padding: 5px;
      float: left;
    }

    .clearfix::after {
      content: "";
      clear: both;
      display: table;
      /* p          */
    }
  </style>
</head>

<body>

  <h2>Images Side by Side</h2>
  <p>Float images side by side</p>

  <div class="clearfix">
    <div class="img-container">
      
    </div>
    <div class="img-container">
      
    </div>
    <div class="img-container">
      
    </div>
  </div>
</body>
</html>
```

```
</div>
```

```
<p>
```

Note that we also use the clearfix hack to take care of the layout flow, and that we add the box-sizing property to make sure that the image container doesn't break due to extra padding. Try to remove this code to see the effect.

```
</p>
```

```
</body>
```

```
</html>
```

Equal Height Boxes

, quick fix

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Equal Height Boxes</title>
  <style>
    * {
      box-sizing: border-box;
    }

    .box {
      float: left;
      width: 50%;
      padding: 50px;
      height: 300px;
    }

    .clearfix::after {
      content: "";
      clear: both;
    }
  </style>
</head>
<body>
  <div class="clearfix">
    <div class="box">
      <img alt="Placeholder for image" data-bbox="100 100 200 200" />
    </div>
    <div class="box">
      <img alt="Placeholder for image" data-bbox="300 100 400 200" />
    </div>
  </div>
</body>
</html>
```

```

        display: table;
    }
</style>
</head>

<body>

<h2>Equal Height Boxes</h2>
<p>Floating boxes with equal heights</p>

<div class="clearfix">
  <div class="box" style="background-color:#bbb">
    <h2>Box01</h2>
    <p>Some content, some content, some content</p>
  </div>
  <div class="box" style="background-color:#ccc">
    <h2>Box02</h2>
    <p>Some content, some content, some content</p>
    <p>Some content, some content, some content</p>
    <p>Some content, some content, some content</p>
  </div>
</div>

<P>
  This example is not very flexible. It is ok to use CSS height if you
  can guarantee that the boxes will always have the same amount of content in
  them, but that's not always the case. If you
  try the example above on a mobile phone (or resize the browser
  window), you will see that the second box's content will be displayed
  outside of the box.
</P>
<p>
  Go back to the tutorial and find another solution, if this is not
  what you want.
</p>
</body>

</html>

```

가
 ,
 가
 가
 CSS3 Flexbox가

```

<!DOCTYPE html>
<html lang="en">

```

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Flexible Boxes</title>
  <style>
    .flex-container {
      display: flex;
      flex-wrap: nowrap;
      background-color: dodgerblue;
    }

    .flex-container .box {
      background-color: #f1f1f1;
      width: 50%;
      margin: 10px;
      text-align: center;
      line-height: 75px;
      font-size: 30px;
    }
  </style>
</head>

<body>

  <div class="flex-container">
    <div class="box">
      Box 01 - This is some text to make sure that the content gets
      really tall. This is some text to make sur that the content gets really
      tall.
    </div>
    <div class="box">
      Box 02 - My height will follow Box 01.
    </div>
  </div>

  <p>Try to resize the browser window to see the flexible layout.</p>
  <p>
    <strong>Note:</strong>Flexbox is not supported in Internet Explorer
    10 or earlier versions.
  </p>

</body>

</html>
```

Navigation Menu

float

가

.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Navigation menu with float property</title>
  <style>
    ul {
      list-style-type: none;
      margin: 0;
      padding: 0;
      overflow: hidden;
      background-color: #333;
    }

    li {
      float: left;
    }

    li a {
      display: inline-block;
      color: yellow;
      text-align: center;
      padding: 14px 16px;
      text-decoration: none;
    }

    li a:hover {
      background-color: red;
    }

    .active {
      background-color: blue;
    }
  </style>
</head>

<body>
  <ul>
    <li><a href="#home" class="active">Home</a></li>
    <li><a href="#news">News</a></li>
```

```
<li><a href="#contact">Contact</a></li>
<li><a href="#about">About</a></li>
</ul>
</body>
</html>
```

Web Layout Example

float

., CSS, float, examples

From: <http://rwiki.repia.com/> -

. - 2023.12

Permanent link:

http://rwiki.repia.com/doku.php?id=wiki:css:css_note:css_float_examples&rev=1615856702 

Last update: **2022/03/10 19:52**