CSS Float Examples

- description : CSS Float Examples
- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-03-15

Source of the article

- CSS Layout - Float Examples

Grid of Boxes / Equal Width Boxes

float속성을 사용하면 콘텐츠 박스들을 쉽게 나란히 배치할 수 있습니다.

예제

```
<!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>Grid of Boxes</title>
    <style>
        * {
            box-sizing: border-box;
        }

        .box {
            float: left;
            width: 33.33%;  /* 3개의 박스, 4개 박스는 25%, 2개 박스는 50% */
            padding: 50px;
        }
```

```
        .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 w also us the clearfix hack to take care of the layout flow, and that 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>
```

3개의 플로팅 박스를 쉽게 나란히 만들 수 있습니다. 하지만 각 박스의 너비를 확대하는 것 (예, 패딩 또는 보더)을 추가할 경우, 박스 배치는 어긋납니다.

''box-sizing'' 속성은 해당 박스의 총 너비 (그리고 높이)에 패딩과 보더를 포함하게 하여, 박스 내부에 패딩이 위치하게 하여 박스 배치가 어긋나지 않습니다.

Images Side By Side

박스의 그리드는 이미지를 나란히 보여주는 것에도 사용할 수 있습니다.

예제

```
<!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;
            /* p를 이미지 하단에 배치 */
        }
    </style>
</head>

<body>

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

    <div class="clearfix">
        <div class="img-container">
            <img src="img_5terre.jpg" alt="Italy" style="width:100%">
        </div>
        <div class="img-container">
            <img src="img_forest.jpg" alt="Forest" style="width:100%">
        </div>
        <div class="img-container">
            <img src="img_mountains.jpg" alt="Mountains" style="width:100%">
        </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 image
container doesn't break due to extra padding. Try
        to remove this code to see the effect.
    </p>

</body>

</html>
```

Equal Height Boxes

이전의 예제에서, 동일한 너비로 박스들을 나란히 배치하는 방법을 확인하였습니다. 그러나, 동일한 높이로 플로팅 박스들을 만드는 것은 쉽지 않습니다.
하지만, quick fix는 고정 높이를 설정하는 것을 아래 예제에서 살펴보겠습니다.

예제

```
<!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;
```

```
            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>
```

하지만, 이렇게 하는 것은 매우 유연하지 않습니다. 박스들 내부에 항상 동일한 양의 콘텐츠가 있다는 것을 보장할 수 있는 경우에는 괜찮습니다.
하지만 많은 경우 콘텐츠가 동일하지 않습니다. 상기 예제를 휴대폰에서 시도하면, 두 번째 박스의 콘텐츠가 박스 외부에 표시되는 것을 확인할 수 있습니다.
자동으로 가장 긴 박스만큼 길게 박스를 늘릴 수 있는 CSS3 Flexbox가 이런 경우에 유용합니다.

예제

```
<!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

하이퍼링크 목록iption : CSS Float Examples

- author : 오션
- email : shlim@repia.com
- lastupdate : 2021-03-15

</WRAP>

Source of the article

- CSS Layout - Float Examples

Grid of Boxes / Equal Width Boxes

float속성을 사용하면 콘텐츠 박스들을 쉽게 나란히 배치할 수 있습니다.

예제

```
<!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>Grid of Boxes</title>
    <style>
        * {
            box-sizing: border-box;
        }

        .box {
            float: left;
            width: 33.33%;  /* 3개의 박스, 4개 박스는 25%, 2개 박스는 50% */
            padding: 50px;
        }
```

```
        .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 w also us the clearfix hack to take care of the layout flow, and that 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>
```

3개의 플로팅 박스를 쉽게 나란히 만들 수 있습니다. 하지만 각 박스의 너비를 확대하는 것 (예, 패딩 또는 보더)을 추가할 경우, 박스 배치는 어긋납니다.

''box-sizing'' 속성은 해당 박스의 총 너비 (그리고 높이)에 패딩과 보더를 포함하게 하여, 박스 내부에 패딩이 위치하게 하여 박스 배치가 어긋나지 않습니다.

Images Side By Side

박스의 그리드는 이미지를 나란히 보여주는 것에도 사용할 수 있습니다.

예제

```
<!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;
            /* p를 이미지 하단에 배치 */
        }
    </style>
</head>

<body>

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

    <div class="clearfix">
        <div class="img-container">
            <img src="img_5terre.jpg" alt="Italy" style="width:100%">
        </div>
        <div class="img-container">
            <img src="img_forest.jpg" alt="Forest" style="width:100%">
        </div>
        <div class="img-container">
            <img src="img_mountains.jpg" alt="Mountains" style="width:100%">
        </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 image
container doesn't break due to extra padding. Try
        to remove this code to see the effect.
    </p>

</body>

</html>
```

Equal Height Boxes

이전의 예제에서, 동일한 너비로 박스들을 나란히 배치하는 방법을 확인하였습니다. 그러나, 동일한 높이로 플로팅 박스들을 만드는 것은 쉽지 않습니다.
하지만, quick fix는 고정 높이를 설정하는 것을 아래 예제에서 살펴보겠습니다.

예제

```
<!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;
```

```
            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>
```

하지만, 이렇게 하는 것은 매우 유연하지 않습니다. 박스들 내부에 항상 동일한 양의 콘텐츠가 있다는 것을 보장할 수 있는 경우에는 괜찮습니다.
하지만 많은 경우 콘텐츠가 동일하지 않습니다. 상기 예제를 휴대폰에서 시도하면, 두 번째 박스의 콘텐츠가 박스 외부에 표시되는 것을 확인할 수 있습니다.
자동으로 가장 긴 박스만큼 길게 박스를 늘릴 수 있는 CSS3 Flexbox가 이런 경우에 유용합니다.

예제

```
<!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

하이퍼링크 목록과 `float`속성을 사용하여 가로 배치 메뉴를 만듭니다.

예제

오션,, CSS, float, examples

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:css:css_note:css_float_examples&rev=1615855323

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