

Examples of classic page layouts

INTRODUCTION

In this section, we will show some "classic" CSS layout techniques for designing an HTML page that uses the new sectioning elements.

We embed examples from this very good post about "Positioning content": <http://learn.shayhowe.com/html-css/positioning-content/>. This is recommended reading as it details how to use the CSS `float` property to layout a Web page.

The examples below are given "as is" to give you some hints. There are lots of other possibilities on using CSS to position elements.

EXAMPLE 1: A `<SECTION>` ON THE LEFT AND AN `<ASIDE>` ON THE RIGHT, USING THE `FLOAT` AND `WIDTH` CSS PROPERTIES

This example uses the following HTML structure:

```
10. <header>
    <code>&lt;header&gt;</code>
    </header>
    <section>
    <code>&lt;section&gt; <br> float: left;</code>
    </section>
    <aside>
    <code>&lt;aside&gt; <br> float: right;</code>
    </aside>
    <footer>
```

```
<code>&lt;footer&gt;</code>  
</footer>
```


Here we use the CSS rule `float:left` for the `<section>` and the CSS rule `float:right` for the `<aside>`. When an element floats, it goes out of the normal flow of the HTML element. Then by default it floats to the edge of its parent; and its size depends on the elements it contains. So, in order to fill the whole horizontal space, we prefer here to "force the width" by setting the CSS `width` property with a percentage. So we took `width: 63%` for the `<section>` on the left and `width: 30%` for the `<aside>` on the right.

You can look at the complete CSS code in the interactive example below (click on the CSS or HTML text in the menu bar below, or click "edit on codepen" to change the code and see the results):

HTML

CSS

Result

Edit on  CODEPEN

<header>

<section>
float: left;

<aside>
float: right;

<footer>

EXAMPLE 2: THREE SECTIONS CENTERED, OF EQUAL SIZE, ALSO USING THE `FLOAT` AND `WIDTH` CSS PROPERTIES

Here we show how to make a 3 column layout using the CSS `float` property.

HTML code:

```
10. <header>
    <code>&lt;header&gt;</code>
  </header>

  <section>
    <code>&lt;section&gt; <br> float: left;</code>
  </section>

  <section>
    <code>&lt;section&gt; <br> float: left;</code>
  </section>

  <section>
    <code>&lt;section&gt; <br> float: left;</code>
  </section>

  <footer>
    <code>&lt;footer&gt;</code>
  </footer>
```

Instead of having one element with a `float:left` and one element with a `float:right` property, we instead use `float:left` for all three of them, and we give a `width:30%` CSS property value to each `<section>`. We also set a small `margin` so that the columns have a gap between them.

Look at the CSS code of the example below:

[HTML](#)[CSS](#)[Result](#)[Edit on](#)  [CODEPEN](#)

EXAMPLE 3: SAME RESULT USING THE CSS `flex` PROPERTY

This example uses the CSS `flex` property to achieve a result similar to the one shown in Example 2.

[HTML](#)[CSS](#)[Result](#)[Edit on](#)  [CODEPEN](#)