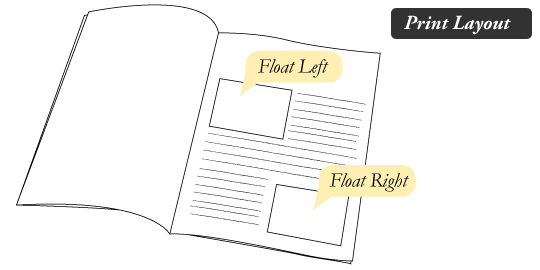
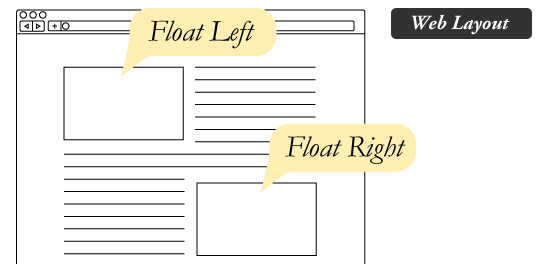
<https://css-tricks.com/all-about-floats/>

Übersetzen Sie den folgenden Text sinngemäß

**Float** is a CSS positioning property. To understand its purpose and origin, we can look to print design. In a print layout, images may be set into the page such that text wraps around them as needed. This is commonly and appropriately called "text wrap". Here is an example of that.



In page layout programs, the boxes that hold the text can be told to honor the text wrap, or to ignore it. Ignoring the text wrap will allow the words to flow right over the image like it wasn't even there. This is the difference between that image being part of the flow of the page (or not). Web design is very similar.



In web design, page elements with the CSS float property applied to them are just like the images in the print layout where the text flows around them. Floated elements remain a part of the flow of the web page. This is distinctly different than page elements that use absolute positioning. Absolutely positioned page elements are **removed** from the flow of the webpage, like when the text box in the print layout was told to ignore the page wrap. Absolutely positioned page elements will not affect the position of other elements and other elements will not affect them, whether they touch each other or not.

Setting the float on an element with CSS happens like this:

#sidebar {

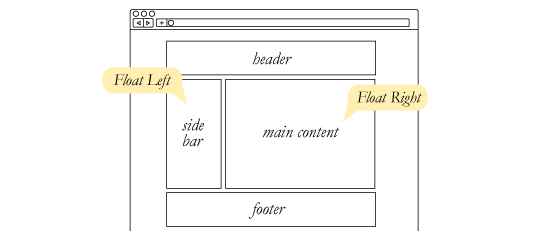
float: right;

}

There are four valid values for the float property. **Left** and **Right** float elements those directions respectively. **None** (the default) ensures the element will not float and **Inherit** which will assume the float value from that elements parent element.

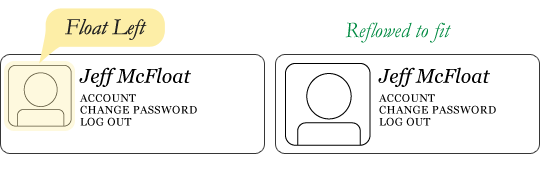
### [#](https://css-tricks.com/all-about-floats/#article-header-id-1)What are floats used for?

Aside from the simple example of wrapping text around images, floats can be used to create **entire web layouts**.

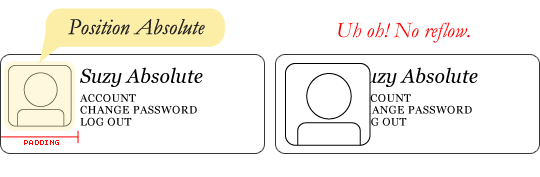


While floats can still be used for layout, these days, we have much stronger tools for creating layout on web pages. Namely, [Flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) and [Grid](https://css-tricks.com/snippets/css/complete-guide-grid/). Floats are still useful to know about because they have some abilities entirely unique to them, which is all covered in this article.

Floats are also helpful for layout in smaller instances. Take for example this little area of a web page. If we use float for our little avatar image, when that image changes size the text in the box will reflow to accommodate:

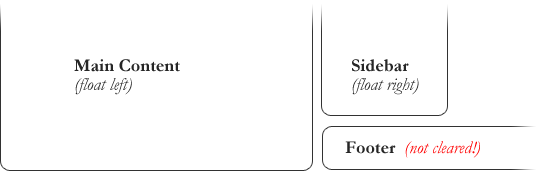


This same layout could be accomplished using relative positioning on container and absolute positioning on the avatar as well. In doing it this way, the text would be unaffected by the avatar and not be able to reflow on a size change.



### [#](https://css-tricks.com/all-about-floats/#article-header-id-2)Clearing the Float

Float's sister property is clear. An element that has the clear property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float. Again an illustration probably does more good than words do.

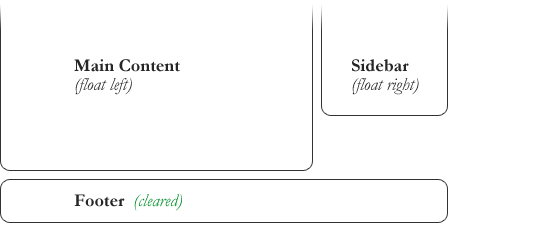


In the above example, the sidebar is floated to the right and is shorter than the main content area. The footer then is required to jump up into that available space as is required by the float. To fix this problem, the footer can be cleared to ensure it stays beneath both floated columns.

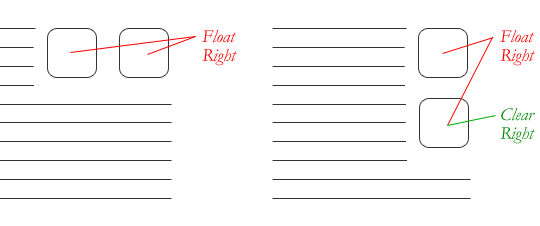
#footer {

clear: both;

}

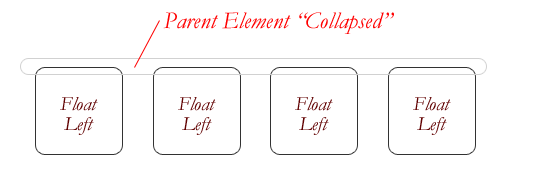


Clear has four valid values as well. **Both** is most commonly used, which clears floats coming from either direction. **Left** and **Right** can be used to only clear the float from one direction respectively. **None** is the default, which is typically unnecessary unless removing a clear value from a cascade. **Inherit** would be the fifth, but is strangely not supported in Internet Explorer. Clearing only the left or right float, while less commonly seen in the wild, definitely has its uses.

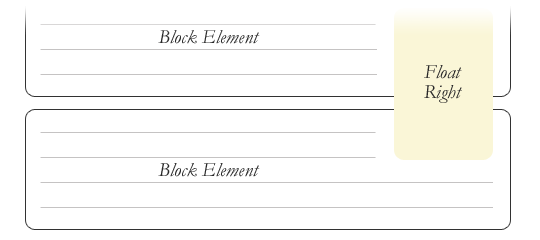


### [#](https://css-tricks.com/all-about-floats/#article-header-id-3)The Great Collapse

One of the more bewildering things about working with floats is how they can affect the element that contains them (their "parent" element). If this parent element contained nothing but floated elements, the height of it would literally collapse to nothing. This isn't always obvious if the parent doesn't contain any visually noticeable background, but it is important to be aware of.



As anti-intuitive as collapsing seems to be, the alternative is worse. Consider this scenario:



If the block element on top were to have automatically expanded to accommodate the floated element, we would have an unnatural spacing break in the flow of text between paragraphs, with no practical way of fixing it. If this were the case, us designers would be complaining much harder about this behavior than we do about collapsing.

Collapsing almost always needs to be dealt with to prevent strange layout and cross-browser problems. We fix it by clearing the float **after** the floated elements in the container but **before** the close of the container.

### [#](https://css-tricks.com/all-about-floats/" \l "article-header-id-7)Video

I did [a screencast](http://css-tricks.com/video-screencasts/42-all-about-floats-screencast/) a while back explaining many of these float concepts.