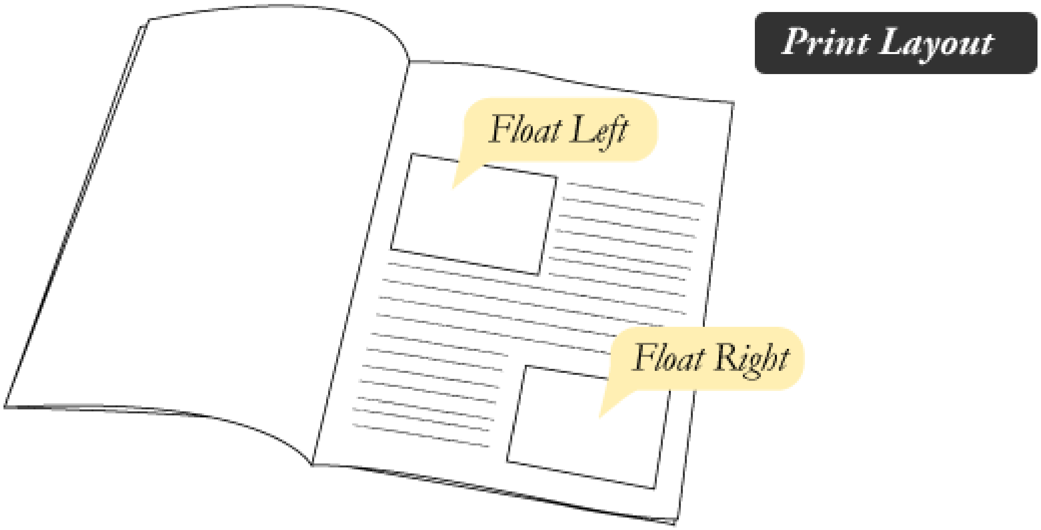
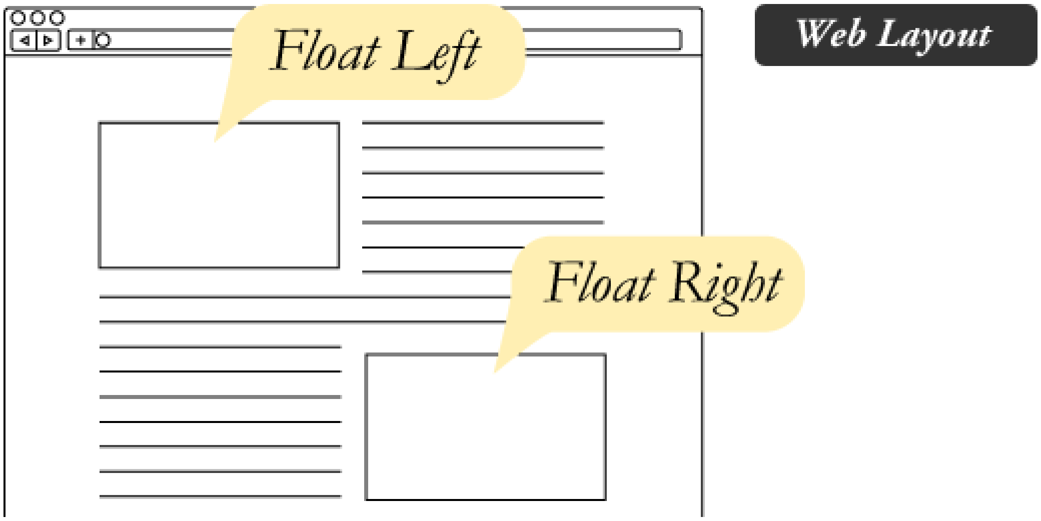
e.g. flexbox

What is "Float"?

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:

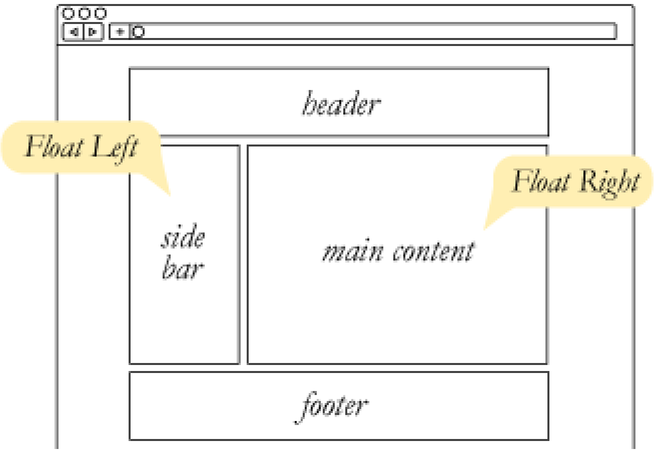
CSS

```css
#sidebar {
  float: right;
}
```

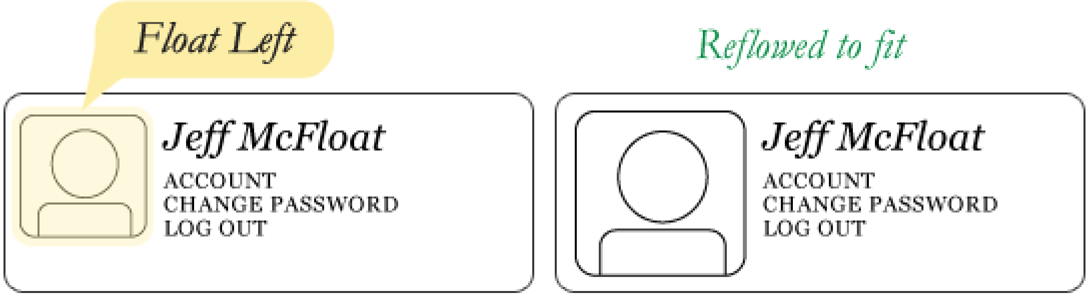
There are four valid values for the float property. **Left** and **Right** float elements those directions respectively. **None** (the default) ensures the element will not float and **Inherit** which will assume the float value from that elements parent element.

What are floats used for?

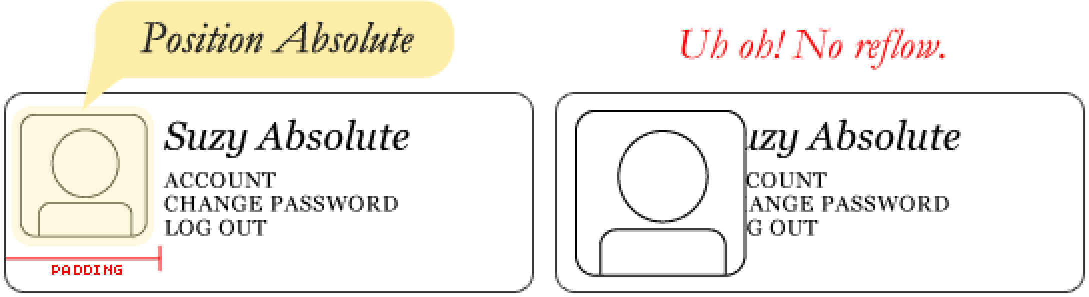
Aside from the simple example of wrapping text around images, floats can be used to create **entire web layouts**.



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:

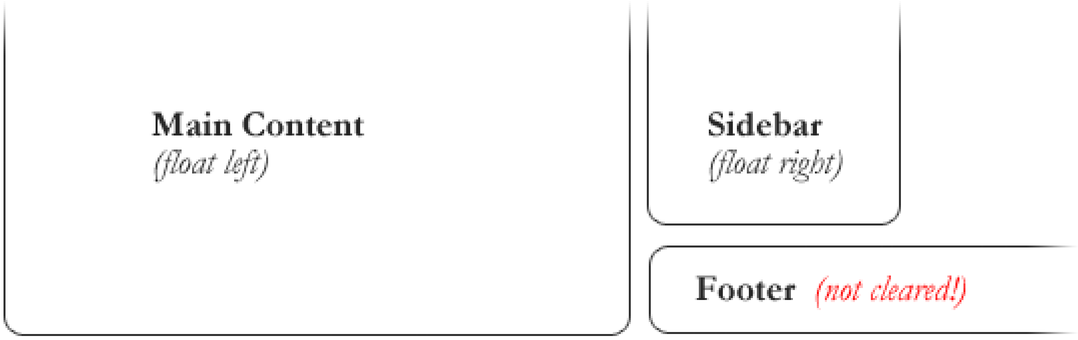


This same layout could be accomplished using relative positioning on container and absolute positioning on the avatar as well. In doing it this way, the text would be unaffected by the avatar and not be able to reflow on a size change.



# Clearing the Float

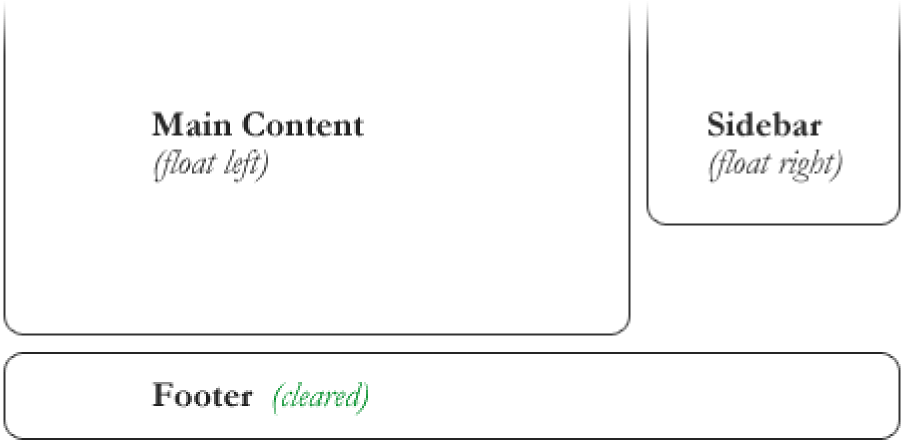
Float's sister property is `clear`. An element that has the clear property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float. Again an illustration probably does more good than words do.



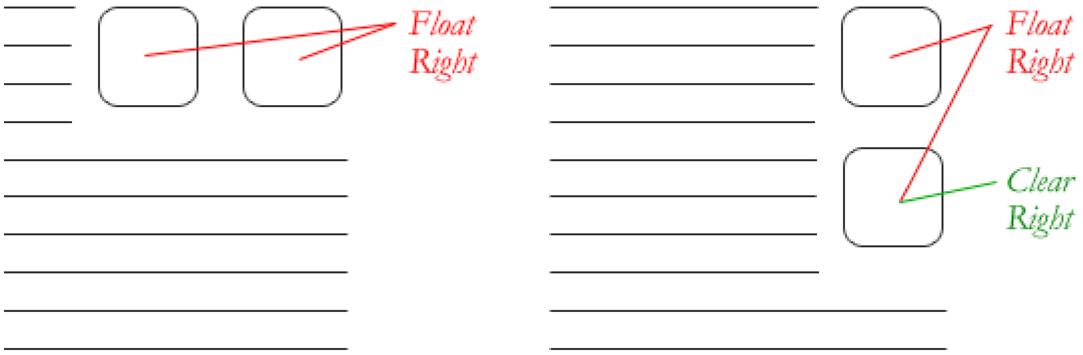
In the above example, the sidebar is floated to the right and is shorter than the main content area. The footer then is required to jump up into that available space as is required by the float. To fix this problem, the footer can be cleared to ensure it stays beneath both floated columns.

CSS

```css
#footer {
  clear: both;
}
```

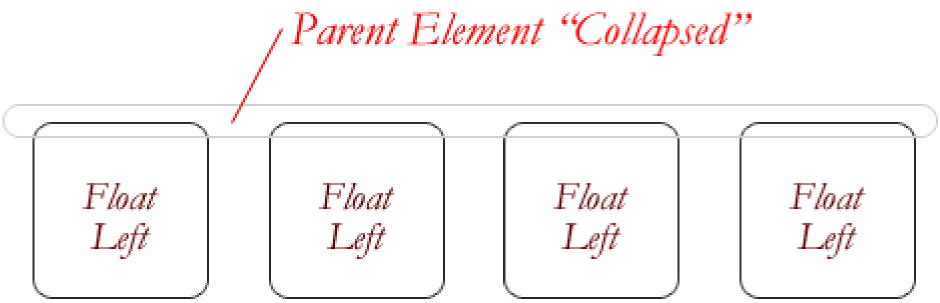


Clear has four valid values as well. **Both** is most commonly used, which clears floats coming from either direction. **Left** and **Right** can be used to only clear the float from one direction respectively. **None** is the default, which is typically unnecessary unless removing a clear value from 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.

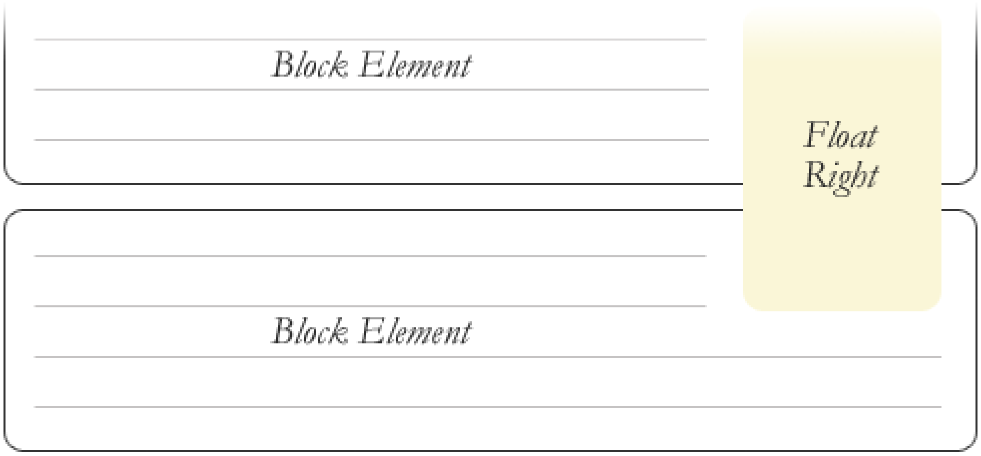


The Great Collapse

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.

Techniques for Clearing Floats

If you are in a situation where you always know what the succeeding element is going to be, you can apply the `clear: both;` value to that element and go about your business. This is ideal as it requires no fancy hacks and no additional elements making it perfectly semantic. Of course things don't typically work out that way and we need to have more float-clearing tools in our toolbox.

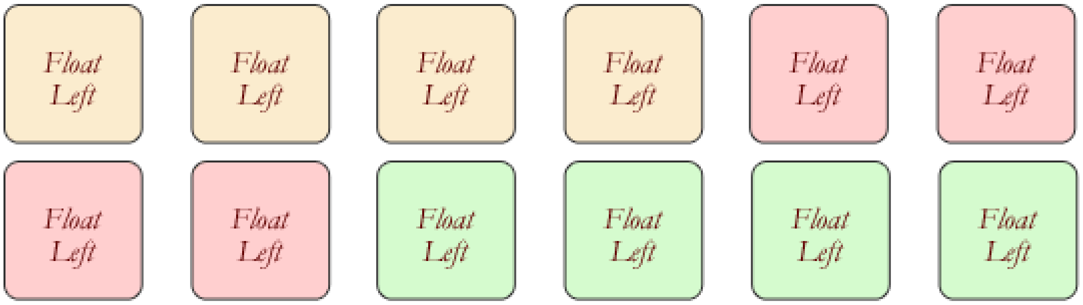
- **The Empty Div Method** is, quite literally, an empty div. `<div style="clear: both;"></div>` . Sometimes you'll see a `<br>` element or some other random element used, but div is the most common because it has no browser default styling, doesn't have any special function, and is unlikely to be generically styled with CSS. This method is scorned by semantic purists since its presence has no contextual meaning at all to the page and is there purely for presentation. Of course in the strictest sense they are right, but it gets the job done right and doesn't hurt anybody.
- **The Overflow Method** relies on setting the overflow CSS property on a parent element. If this property is set to auto or hidden on the parent element, the parent will expand to contain the floats, effectively clearing it for succeeding elements. This method can be beautifully semantic as it may not require an additional elements. However if you find yourself adding a new div just to apply this, it is equally as non-semantic as the empty div method and less adaptable. Also bear in mind that the overflow property isn't specifically for clearing floats. Be careful not to hide content or trigger unwanted scrollbars.
- **The Easy Clearing Method** uses a clever CSS pseudo selector (`:after`) to clear floats. Rather than setting the overflow on the parent, you apply an additional class like "clearfix" to it. Then apply this CSS:

CSS

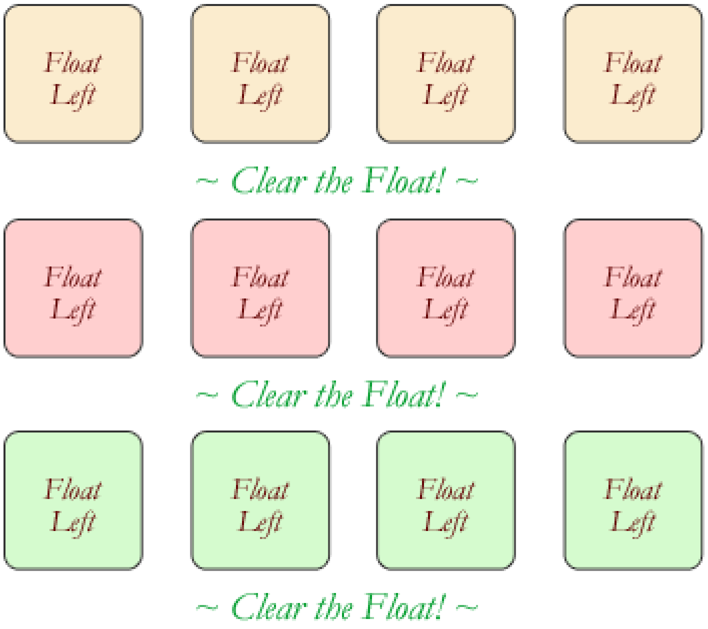
```css
.clearfix:after {
   content: ".";
   visibility: hidden;
   display: block;
   height: 0;
   clear: both;
}
```

This will apply a small bit of content, hidden from view, after the parent element which clears the float. This isn't quite the whole story, as additional code needs to be used to accomodate for older browsers.

Different scenarios call for different float clearing methods. Take for example a grid of blocks, each of different types.



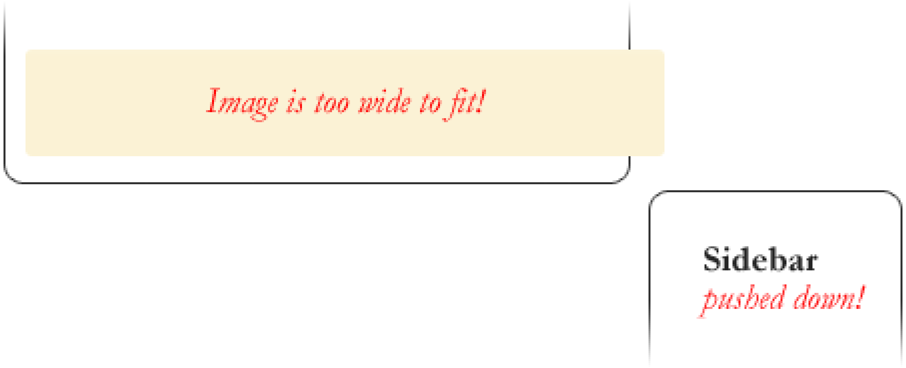
To better visually connect the similar blocks, we want to start a new row as we please, in this case when the color changes. We could use either the overflow or easy clearing method if each of the color groups had a parent element. Or, we use the empty div method in between each group. Three wrapping divs that didn't previously exist or three after divs that didn't previously exist. I'll let you decide which is better.



Problems with Floats

Floats often get beat on for being *fragile*. The majority of this fragility comes from IE 6 and the slew of float-related bugs it has. As more and more designers are dropping support for IE 6, you may not care, but for the folks that do care here is a quick rundown.

- **Pushdown** is a symptom of an element inside a floated item being *wider than the float itself* (typically an image). Most browsers will render the image outside the float, but not have the part sticking out affect other layout. IE will expand the float to contain the image, often drastically affecting layout. A common example is an image sticking out of the main content push the sidebar down below.

  Quick fix: Make sure you don't have any images that do this, use `overflow: hidden` to cut off excess.

- **Double Margin Bug** - Another thing to remember when dealing with IE 6 is that if you apply a margin in the same direction as the float, it will double the margin. *Quick fix:* set `display:`

`inline` on the float, and don't worry it will remain a block-level element.

- The **3px Jog** is when text that is up next to a floated element is mysteriously kicked away by 3px like a weird forcefield around the float. *Quick fix:* set a width or height on the affected text.
- In IE 7, the **Bottom Margin Bug** is when if a floated parent has floated children inside it, bottom margin on those children is ignored by the parent. *Quick fix:* using bottom padding on the parent instead.

Alternatives

If you need text wrapping around images, there really aren't any alternatives for float. Speaking of which, check out this rather clever technique for wrapping text around irregular shapes. But for page layout, there definitely are choices. Eric Sol right here on A List Apart has an article on Faux Absolute Positioning, which is a very interesting technique that in many ways combines the flexibility of floats with the strength of absolute positioning. CSS3 has the Template Layout Module that, when widely adopted, will prove to be the page layout technique of choice.

Video

I did a screencast a while back explaining many of these float concepts.

Comments

V1

# JUNE 30, 2008

http://www.quirksmode.org/css/clearing.html

That is what i use for clearning my floats. hotness.

Bogdan Pop

# JULY 9, 2009

Quoted from the very site you linked above:
"# Some browsers also need a width or height for the container div."

What do you do when the content is dynamic? Bummer. That won't work…

Thomas Offinga

# JUNE 30, 2008

Great job on this! You should explain things in this style more often, it's really refreshing and easy to follow! :)

koew

# JULY 9, 2009

Agreed. You really impressed me with this article, Chris. The examples you've drawn are simple, easy to understand and beautiful to look at.

You should continue with articles like these, Chris! Much kudos!

Epic Alex

# JUNE 30, 2008

Hi Chris, just a question about where you say:

> <BLOCKQUOTE>
>
> In fact, if the parent element contains only floated elements, browsers will render the height at zero. – If you clear the float before the closing tag of the parent element, you can fix this.
>
> </BLOCKQUOTE>

I've always resolved that by floating the parent element as well, is this bad practice, even though it achieves the same thing? Plus that doesn't add non-semantic markup.

Just wondered what your thoughts were?

Mars

# JULY 8, 2009

Floating the parent div is the easiest and cleanest method to deal with the collapse. A floated div with a specified width, will expand vertically to contain its child elements.

emonweb

# JULY 8, 2009

if all children are floated, set overflow: auto to parent element and see the effect. it works most of the modern browsers.

Chris Coyier

# JUNE 30, 2008

@Epic Alex: But what if you don't want the parent element floated? That could cause layout problems of it's own. If you use the clearfix method or the overflow: auto; method, you can also clear the float fixing the parent element size without adding extra markup.

Mat

# JUNE 30, 2008

This is my “Ah-Ha!” moment… haha.

liam

# JUNE 30, 2008

Wow, this is sweet. I think the way you’ve presented the information is great, definitely would love to see more set out like this. Nice read.

Brent Traut

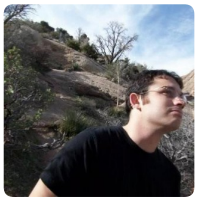
# JUNE 30, 2008

Chris,

I discovered a very relevant link in the last few days concerning clearfix. I suggest you check it out and possibly even apply its recommendations to this article:

http://perishablepress.com/press/2008/02/05/lessons-learned-concerning-the-clearfix-css-hack/

Brent



Tim Wright

# JUNE 30, 2008

I think that’s the most comprehensive explanation of floats I’ve ever seen, very nice

Marco

# JUNE 30, 2008

Now that’s what I would really love to call one great article, explaining everything even ’till the minor details.

Keep up the great work.

Matt

# JUNE 30, 2008

I can’t believe all these years I had a completely different perception of how “clear” worked and could never figure out why it worked sometimes but not others. It’s all perfectly clear (bad pun) now. Thank you so much for this article.

Jason

# JUNE 30, 2008

Unfortunately, I learned this the hard way years ago! Good article, thanks!

Anders

# JUNE 30, 2008

Nice summary Chris!

Would love to see more of these in the future..

> <BLOCKQUOTE>
>
> Elements which contain floated elements do not calculate their height properly. In fact, if the parent element contains only floated elements, browsers will render the height at zero.
>
> </BLOCKQUOTE>

This is actually the correct behaviour according to the specs, so it’s not a matter of the browser getting it wrong. Here’s an article illustrating why this makes sense.

Brandon

# JUNE 30, 2008

Instead of adding more divs to my markup, I simply use <br />inside of my parent div, then use

```
.parent br { clear: both; }
```

in my css to make the parent div clear and expand to the height of the internal floats. It has the same effect with less markup and div confusion.

Gary

# JUNE 30, 2008

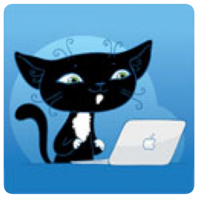
Good article. Floats are great, but they can be aggravating at times.

The article on faux absolute positioning you mention is, as you said, interesting. I played with it and really liked it. At least until I read this argument against faux absolute positioning.

Phillip

# JUNE 30, 2008

Awesome awesome awesome post. I think if there is one thing I use the most it's "float". I actually didn't know about some of these bugs. Thanks!



jbj

# JUNE 30, 2008

An excellent article, really. I enjoyed reading it.
Greetings from Belgium!

Jeff Starr

# JUNE 30, 2008

Thanks for the ping Chris — I updated my Clearfix article to reflect the Firefox-3 fix presented in your post. Now I'm off to update a gazillion different stylesheets!

Btw, I am fairly certain that the "official" abbreviation for Firefox is "Fx" and not "FF", for whatever it's worth..

Eric Wendelin

# JUNE 30, 2008

Just when I thought I knew a lot about floats…. thanks Chris!

guest

# JUNE 30, 2008

Something OT:
Basicly I don’t think the “float” property is tend to be used as a layout property,
but as a typography property maybe â€” it’s coming from the “align” property of a IMG tag.

There is no such a property specificly tend to be used for layout yet in css2.1 , you say “position”? well, position is just so lack of control.

CSS2.1 is still basicly a document decorating language, same as CSS1, complex layout is far out of it’s range. Every CSS technic now we are using for layout, is a kind of hack, including float and position. What about table layout? Yes it’s obviously a hack, too.

So, we are hacking everything here.

Fouad Masoud

# JULY 1, 2008

> <BLOCKQUOTE>
>
> Blockquote Overflow: auto; on the parent element
>
> </BLOCKQUOTE>

overflow: hidden; works exactly the same as overflow: auto; it clears the float and if you specify a width it makes sure that the parent element hides any thing that goes out the original width. With width specified you can fix a couple of IE6/7 bugs where margin bottom of the last element doesn’t work.

Generally speaking and i think everybody here knows this. specifying a width to elements solves like 70% of IE rendering bugs.

@Jeff Starr it is Fx they used it on they site for a while, i cant find it there anymore though.

this is a link to whats new with Fx support to CSS http://tinyurl.com/yuxlbs

Chris Spooner

# JULY 1, 2008

Useful article to have saved, picked up some good tips too!

Olivier J

# JULY 1, 2008

I don’t want to sound rude but your article did sound a little bit confusing to me, not “clear” enough I would say:)

Here is a previous article I had read which explains the same points in a much simple and complete way.

http://kilianvalkhof.com/2008/css-xhtml/understanding-css-positioning-part-1/
http://kilianvalkhof.com/2008/css-xhtml/understanding-css-positioning-part-2/

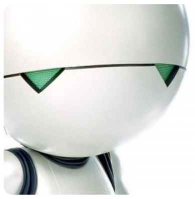
Jeff Starr

# JULY 1, 2008

@Fouad Masoud:

http://www.mozilla.com/en-US/firefox/releases/1.5.html#FAQ

Check #8..



Travis

# JULY 1, 2008

I think this is a very detail coverage on float usage! I believe you can never have enough wrappers to make sure no surprises from your floating elements.

Sean

# JULY 1, 2008

Thank you for a well written article. It really helped clear up some of the confusion I had re: floating.

Paul B.

# AUGUST 3, 2008

I can't think of a single thing that's caused me more woes than floats. When they work, they *work*, but getting to that point is full of all kinds of trickery. This is a great guide that I'll definitely bookmark for the next time I need to solve a float problem.

Walkman69

# OCTOBER 10, 2008

This is a **really great** article. Actually it's the best information i have stepped over since the time i learnt about how to use *Favlets* and when i came in contact with http://browsershots.org/.. Look into thoose fenomena if you don't already know about'em =).. They are almost as great as your page.. ^^

Marilyn

# OCTOBER 30, 2008

Thank you for this post… you've helped me stop pulling out my hair!!

Pete White

# DECEMBER 22, 2008

Great article, its amazing all the little quirks between browsers. I've spent a lot of time fixing little CSS issues!

Sai-Kit Hui

# JULY 8, 2009

Good article, but I still prefer to avoid floats at all costs. My blog is built using inline-block and it is float-free. It's not a perfect alternative, but I prefer it over floats anytime.

Check out <a href="my webdev section on how to use inline-block.

Russell Bishop

# JULY 8, 2009

Your empty div method should be tidied up a little;

Firstly, it should be written as

And then the css;

.clear { width: 0; height: 0; overflow: hidden; clear: both; }

This cuts out inline styles, and also an empty div (the space needs to be in there).

GraphicGorilla

# JULY 8, 2009

reshaping my mind about css every 5 articles. What I'd give to get just half an hour with you. Getting the CSS right over some good Belgian beers.

Robert DeBoer

# JULY 8, 2009

Sweet article. I always enjoy it when people take the time to actually explain why and how something works.

If I understand how and why something works in the first place, I am more apt to use it correctly and produce better CSS quicker with fewer bugs and hacks.

Robert DeBoer

# JULY 8, 2009

More tutorials like this would be great – such as an explanation of tables – all their parts (thead, summary, tbody, etc) and behaviors.

TheDoc

# JULY 8, 2009

Love the title, I chuckled whether intentional or not haha

TheDoc

# JULY 8, 2009

I meant the “Sub-title”, “What is Float?”

Pok3

# JULY 8, 2009

Great article as aways. :D
BTW, Chris. I think you need to add those social-bookmarking buttons again. (I digged this one >>> http://digg.com/d1w748)

miziology

# JULY 8, 2009

this is what im looking for..float bring me headache but now i know how..thanks chris

RJ

# JULY 8, 2009

This is why I come to this site everyday and this is why I am a better designer because of articles like this. Great post for beginner designers appreciate all the work you put out on this site.

G.

# JULY 8, 2009

Excellent! This is what I'm looking for. Thank you very much!

JT

# JULY 8, 2009

Thank you, thank you, thank you… this is an *extremely helpful* article.

Karl Oakes

# JULY 9, 2009

Thanks Chris, an excellent article on a subject that can cause everybody problems at some point.

Michael Angrave

# JULY 9, 2009

Thanks so much for this post. You've covered several little areas in here, which although I was aware of, perhaps didn't understand as well as I should.

Your explanations are clear and have helped my understanding.

Cheers.

Matthew Johnson

# JULY 9, 2009

Awesome as always Chris, and I definitely dig the illustrations. Nice work! :D

andrea

# JULY 9, 2009

That’s a must have article. i hope i will remember of it, when after hours of playing around with firebug, my eyes are going in circle.

well done!

Daquan Wright

# JULY 9, 2009

Great run down on float properties Chris. I generally experimented with them when working with wordpress or my latest layouts. Quite simple in concept and it can save you a bit of work instead of positioning things all the time.

Athena Emily

# JULY 10, 2009

The Empty Div Method is my preferred way but sometimes… just sometimes it doesn’t pass the w3 validator.

Nintensity

# JULY 10, 2009

Wow.. Great source of information, very detailed, and very informative not only for those who are new to floats, but for experts like me, its still important to know the basics again. awesome

vbdietz

# JULY 10, 2009

Nicely done tutorial. Though I knew most of it, I did learn one new technique, that of overflow: auto or hidden. I usually did either the empty div or the clearfix:after method.

Thanks for taking the time to write it up in this manner. Its clarity makes it outstanding.

Disslocated

# JULY 11, 2009

For what it’s worth, whenever I use an ’empty’ clearing div, I put a comment (like <!– ie sucks –>) between the opening and closing div tags. I forget the specifics, but an completely empty div caused a problem with a version of IE.

chris

# JULY 11, 2009

There is a very simple solution to all this. use ” display:table” for the parent DIV , and then “display:table-cell” for all child DIVS, unfortunately only firefox, safari and IE8 support this. The minute it becomes available for all browsers, the minute float will be gone for formatting site regions. Floats are best used for inline elements

Allen Baranov

# JULY 13, 2009

Brilliant… where were you a few years ago when I needed to know about this stuff? :)

David - NWD

# JULY 13, 2009

Even though I’ve been using CSS for years, your blog posts always shed new light and uncover things I just did not know.

Impressive explanation!

Thanks Chris!

CJ

# JULY 14, 2009

Have you ever experienced issues with links jumping inside floated div containers?

Rockstar79

# JULY 15, 2009

Great post! I’ve read several articles and books that claim to ‘explain’ floats and their use, but this blows them ALL out of the water.

I also really liked the clean, easy-to-understand pictures. They just help to really get a feel for how the floats work in context.

Thanks for this incredibly useful post!

Web Host Right

# JULY 16, 2009

Thanks, it’s very helpful including the images in your examples to get a clear idea of what is meant to happen.

bineet

# JULY 17, 2009

i am a web designer beginner . we face a lot of problem with float property in css coding .when we give a margin to any container IE make double margin to this container.so to fix this problem i use _margin only for IE. now our headache has over . your example are easy and simple. nice………………..

tim

#JULY 17, 2009

Great article Chris. I only wish this existed when I first started! Keep it up.

Neel

#JULY 17, 2009

Excellent Article, I also positioned div using float on most of the cases as it helped other than using position.

Kat

#JULY 21, 2009

Thanks! This helped me fix a prob in IE6! Yay! You are my hero!

Neel

#JULY 22, 2009

Excellent Article. I am having some problem with the float: but after reading this great article all are gone.

cardeo

#JULY 23, 2009

Great article, I posted a link to it on my blog, I’m constantly dealing with the collapse issue

mike

#AUGUST 16, 2009

Great article. Floats have been one of the most frustrating parts of designing my site. Thanks for clearing it up.

Love the style and diagrams of this article. Please use this format for more articles.

Marc Rogall

#OCTOBER 9, 2013

Hi Chris,

Thanks for this post, I often refer to you for answers to my CSS.

Question: You mentioned “Floated elements remain a part of the flow of the web page.” But aren’t they removed from document flow? Isn’t that why we have to apply a clearfix to parent containers so they don’t collapse?

Chris Coyier

#OCTOBER 9, 2013

Collapsing is an issue, but the floated elements aren’t technically removed from the flow. Imagine how if a floated element was 100px wide, and text flowed around it, then it became 200px wide, the text would move over and flow around that. Still affects other elements == still in flow.

Osama

#JANUARY 2, 2014

**“Floated elements remain a part of the flow of the web page.”
**
Is this info is true ?

Pratyush

#FEBRUARY 18, 2014

Hello Chris,

If you don’t have height and width to the div and you have floated children inside it and you want parent div should take height of child elements. What will be the approach? I know we can give float left to the parent or can make overflow hidden to the parent.

Pratyush

#FEBRUARY 18, 2014

One more question Chris,

I have also read that floated elements are removed from the html flow.

“Imagine how if a floated element was 100px wide, and text flowed around it, then it became 200px wide, the text would move over and flow around that. Still affects other elements == still in flow.”

What does this mean?

Christiano G. de Araújo

#FEBRUARY 19, 2014

if you put a large widht table in main content we have a problem: table extrapolate the div’s width. Anybody know how solve this? With large widht table, withour div´s overflow?

Esol Esek

#JUNE 24, 2014

Only a group of engineering idiots could have removed the center element from web design. Only stupid engineers could’ve made everyone’s lives so fing difficult.

This comment thread is closed. If you have important information to share, please contact us.