

# CSS block elements on a line

Asked 16 years, 1 month ago   Modified 16 years, 1 month ago

Viewed 37k times



8



What's the most common way to deal with a series of block elements that need to be on a line (if javascript needs to be able to modify their widths, for example)?  
What are the pros and cons to applying float:left to each of them vs. using positioning to place them?



html

css

web-standards



Share

Improve this question

Follow

asked Nov 21, 2008 at 18:52



Dan Monego

10.1k ● 6 ● 40 ● 77

5 Answers

Sorted by:

Highest score (default)



10



Well, if you're not too concerned with older browsers (I'm looking at you, IE6) the best way here is to go with

```
display:inline-block;
```



Basically, it creates a box-model element without clearing before or after it, so it remains in the line. Every modern



browser interprets it well.

Share Improve this answer

answered Nov 21, 2008 at 22:56

Follow



Gabe

1,288 ● 14 ● 39

---

try firefox.. it also has issues with this property.. if I'm not mistake there is a -moz-inline-block or something for this..

– [Davy Landman](#) Nov 21, 2008 at 23:02

---

Just today I did some stuff using it (FF3) and had no problems whatsoever, looked he same in FF, Opera, Chrome and IE7 – [Gabe](#) Nov 21, 2008 at 23:04

---

Bad idea, there are simple ways to implement what he's talking about without using sketchy elements that aren't supported in many browsers (IE6 is still a major browser despite how we all feel about it) – [Andrew G. Johnson](#) Nov 21, 2008 at 23:04

---

IE6 can supoort it correctly with minor adaptations ([quirksmode.org/css/display.html](http://quirksmode.org/css/display.html)), but as many things in webdesign, the way to chose to solve a problem is directly related to the public of the site. That's why I said that he could use it if older browsers don't concern him. – [Gabe](#) Nov 21, 2008 at 23:14

---



5



Floating would be my choice, but it really depends on what you wish to achieve. If you can provide a more specific example I would be able to give you a clear reason as to what and why I think you should use.

However here is a brief set of pros and cons that I have come accross (I'm assuming that by positioning you mean absolute positioning):



## Positioning pros:



- very precise positioning in relation to the next ancestor marked as position relative - allows for great flexibility
- allows for elements to be in a different order visually than they are in the DOM

## Positioning cons:

- Harder to line up with other elements since the positioned element is no longer in the document flow and also because of the level of precision required.
- Other elements ignore the absolutely positioned element which means you could have a potential overlap, unless you account for the minimum and maximum size of the positioned element
- harder to implement a footer if you are using absolute positioning for your columns.

## Float pros:

- really easy to construct simple and advanced layouts
- no footer problem
- no worrying about precision, browser takes care of that for you
- parent container stretches

## Float cons:

- Lots of pitfalls for those less experienced with float layouts that may lead to many questions being asked on SO :)

As to the clear:both element that Sebastian mentioned, There's a simple way around that. Lets say you have a container div and 2 floated divs inside.

Html:

```
<div id="con">
  <div class="float"></div>
  <div class="float"></div>
</div>
```

CSS:

```
#con { background:#f0f; }
.float { float:left; width:100px; height:100px; backgr
```

if you were to run this code you would notice that the container div (the magenta coloured one) is only a single pixel high whereas the floated divs were correct - which is the problem Sebastian mentioned. Now you could take his advice and add a br or float the container which would not be very semantic - so here is a slightly more elegant solution. Just add overflow:hidden; to the container div like so:

```
#con { background:#f0f; overflow:hidden; }
```

Now your container should wrap the floated divs properly.

Share Improve this answer

answered Nov 21, 2008 at 22:32

Follow



[Darko](#)

38.8k ● 15 ● 83 ● 107



1

The parent container does not stretch with them unless it is also assigned a float tag or there is a br with clear:both; at the bottom.



I would go with the float:left instead of the positioning. The browser does all the aligning when one object stretches. So there is less for you to care about.



Share Improve this answer

answered Nov 21, 2008 at 18:58

Follow



[Sebastian Hoitz](#)

9,373 ● 13 ● 63 ● 77



1

I think i wouldn't explicitly position the elements but rather instruct the browser to use an inline layout for the elements using display:inline and let the browser handle the positioning.



regarding float vs positioning i think the only way to line them up using positioning is by using absolute positioning, and that means you need to handle re-sizes(of the browser view port) in order to keep the elements in place.



I think that by using the float property the browser handles the re-sizing issues for you and re-renders the element in the correct place.

Share Improve this answer

edited Nov 21, 2008 at 19:57

Follow

answered Nov 21, 2008 at 19:13



[rhinof](#)

51 ● 3

---

The problem with display:inline is it precludes a lot of options for scripting, including dynamically resizing width.

– [Dan Monego](#) Nov 21, 2008 at 22:07

---



0



Only disadvantage to float in situations like this for me has been that you'll either need to left justify them or right justify them -- centering is not an option. However you've mentioned you're using absolute values for widths, so you could just nest all the floated elements in a DIV element and add either margin-right or margin-left to the parent DIV to simulate center alignment.

Share Improve this answer

answered Nov 21, 2008 at 22:34

Follow



[Andrew G. Johnson](#)

27k ● 30 ● 92 ● 137

---