Floats

Floated element looks inline but behaves like a block element so you need to give it an explicit width.

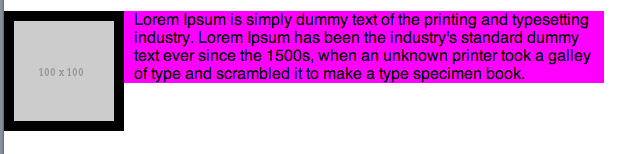
You can put them side by side

For stuck floats – use Clear

Clear has a left, right and \_\_\_ properties

Because a float behaves like a block element, you can apply box model properties to it. You can add margin and padding to modify things.

Parent collapse:



We want the text to be fully with the image.

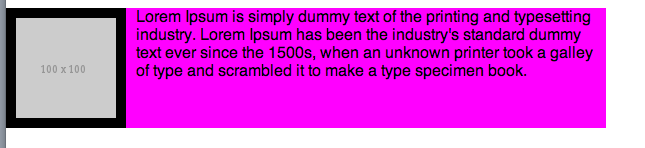
Don’t add content to fix your layout!

/\* Option A - Manually set dimensions \*/

article {

height: 120px;

}



Try NOT to set an explicit height! If we add additional text, then it would go beyond this explicitly set 120px height. Bad.

/\* Option B - float parent \*/

article {

float: right;

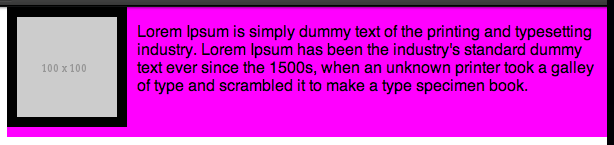
}

If the child is floated, you can also float the parent. The image is the child.

“You left your mom and dad to go to college… and they followed you. That’s essentially what happened here. When you float an element, you’re leaving it, but the container follows you.”

Pro: it works

Con: it can shift the entire layout



/\* Option C - overflow \*/

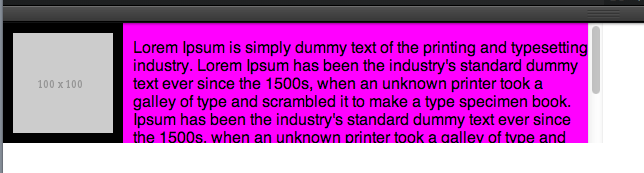
article {

overflow: auto;

}

*auto* – adjusts accordingly (adds scroll bars)

*hidden* – the excess content is hidden



pro: it works

con: it brings in nasty scroll bars – don’t put scroll bars in scroll bars!

They’re not meant for fixing a collapsed parent. It’s meant for clipping something.

/\* Option D - clear:both on very, last child\*/

.clearfix {

clear: both;

}

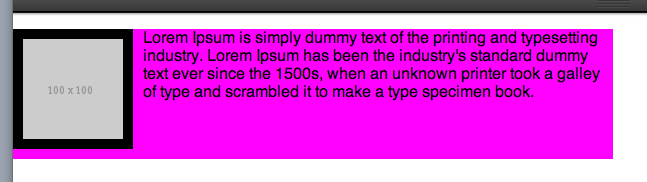
can use *right, left,*  or *both*

Used quite a bit even on the 960 grid system. People love and hate the 960 grid system. If you don’t want to use it, don’t!

<div class=“clearfix”></div> - You’re giving it markup but no content, not the best practice.

Making up an empty piece of content (like a fake child) to pull the “parent” down.

You create a fake class – commonly referred to as clearfix



Grabs the last piece of content and makes sure nothing is floated to the left or the right.

Every time you’d want a non-collapsed parent, you’d have to attach a clearfix class to EVERY SINGLE DIV. Pain in the ass. If you don’t, it will go along with the *document flow* and you’ll end up with this clusterf\*\*\* because it sees space, so it’s going to fill it.



<http://coding.smashingmagazine.com/2009/10/19/the-mystery-of-css-float-property/>

Use this as a reference. It’s the most elegant system that modern grid systems use to deal with this issue of a collapsed parent. It used the after pseudo class.

Can allow you to add content but this is bad because HTML is for content, not CSS.

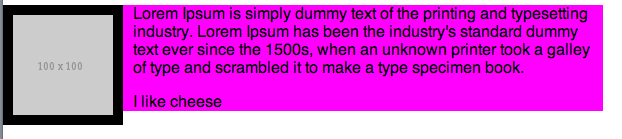
Ie:

/\* Option E - :after \*/

article:after {

content: "I like cheese";

}



this is “pseudo content” – it isn’t real content. Yes it’s there, but google wouldn’t see it when it’s looking at the HTML.

But we can leverage this as a hack.

/\* Option E - :after \*/

article:after {

content: ".";

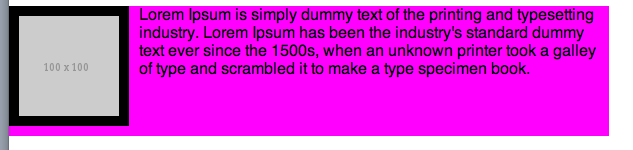
display: block;

clear: both;

height: 0;

visibility: hidden;

}



anytime we have an article, the collapsed parent will fix itself.

But we can modify it to where we give the article a class of “clearfix”

Remember floated elements need an explicit width.

So it becomes:

HTML 🡪 <article class="clearfix">

CSS🡪

/\* Option E - :after \*/

.clearfix:after {

content: ".";

display: block;

clear: both;

height: 0;

visibility: hidden;

}

