**Positioning Within a Colored DIV Box: Child DIV Tags**

Putting text, images, or other HTML content directly into a positioned and colored DIV box produces results which are inconsistent from browser to browser and not entirely satisfying, aesthetically. It is impossible to control the distance between the content and the edge of the DIV box consistently when the content is plopped directly into a colored and positioned DIV box.

To produce consistent appearance and to control properly the distance between the edge of positioned content and the edge of a DIV box, one must put HTML content into positioned DIV tags which are NESTED INSIDE the positioned, colored DIV box.

Example (abbreviated):

<div id="parentDiv">

<div id="childDiv">

<p>Here is some text. And some more text. And so on and so on and so on.</p>

</div>

</div>

Example (in context using an embedded style sheet):

<html>

<head>

<title>Example CSS-P Page</title>

<style type="text/css">

#parentDiv {

position:absolute;

left:100px;

top:30px;

width:200px;

height:200px;

background-color:#CC99CC;

border:solid 1px #CC99CC;

}

#childDiv {

position:absolute;

left:10px;

top:10px;

width:180px;

}

</style>

</head>

<body>

<div id="parentDiv">

<div id="childDiv">

<p>Here is some text. And some more text. And so on and so on and so on.</p>

</div>

</div>

</body>

</html>

Here is the above [example](http://fog.ccsf.edu/~srubin/divchilde1.html) displayed.

In the above example, you can see that I have created a DIV box named "parentDiv" which is 200x200. Inside the "parentDiv" DIV tag, I have nested another DIV tag called "childDiv" which contains a paragraph of text. "childDiv" has been set to be 180 pixels wide.

You will notice that the "childDiv" DIV tag's positioning is set at 10,10. Child nested elements are positioned in relation to the PARENT element, NOT to the web page as a whole. Therefore, "childDiv" is 10 pixels from the left and 10 pixels from the top of the PARENT element's upper-left-hand corner.

The "childDiv" DIV tag has been set to be 180 pixels wide. Since it has been positioned 10 pixels from the left of the parent element, and the parent element is 200 pixels wide, I want to create an even "margin" (it's not actually a CSS margin, but, rather, an artificial margin that we've created); to do this, I have made the "childDiv" 180 pixels wide, which wraps the text 10 pixels from the right edge of the parent box.

The point here is that, because padding, border, and margin properties do not work consistently cross-browser in CSS-P, I have had to create a fake margin by positioning a child element absolutely within an absolutely positioned parent element.

By making parent/child positioned elements in this fashion, you actually end up saving yourself trouble when you start moving into DHTML coding. Thus, if you write only to minimally formatted positioned child elements within a larger parent DIV box, the parent DIV box itself does NOT lose it's appearance characteristics and you can reduce the negative effects of this irritating bug.

If I were to then move "parentDiv", the "childDiv" DIV tag's positioning information would NOT have to be changed, but would move with the parent element automatically.

Example (abbreviated):

#parentDiv {

position:absolute;

left:200px;

top:100px;

width:200px;

height:200px;

background-color:#CC99CC;

border:solid 1px #CC99CC;

}

#childDiv {

position:absolute;

left:10px;

top:10px;

width:180px;

}

Example (in context using an embedded style sheet):

<html>

<head>

<title>Example CSS-P Page</title>

<style type="text/css">

#parentDiv {

position:absolute;

left:200px;

top:100px;

width:200px;

height:200px;

background-color:#CC99CC;

border:solid 1px #CC99CC;

}

#childDiv {

position:absolute;

left:10px;

top:10px;

width:180px;

}

</style>

</head>

<body>

<div id="parentDiv">

<div id="childDiv">

<p>Here is some text. And some more text. And so on and so on and so on.</p>

</div>

</div>

</body>

</html>

Here is the above [example](http://fog.ccsf.edu/~srubin/divchilde2.html) displayed.

Again, notice that although the "parentDiv" positioning information has changed, moving the entire element to another location on the page, the "childDiv" positioning information has NOT changed. This feature allows you to move the parent element around on different HTML pages without having to recalculate the child positions, as well.

You can position more than one element within a parent DIV tag. In fact, you can position as many elements as you like within a parent DIV tag.

Example (in context using an embedded style sheet):

<html>

<head>

<title>Example CSS-P Page</title>

<style type="text/css">

/\* parent DIV tag \*/

#donaldDuck {

position:absolute;

left:30px;

top:75px;

width:300px;

height:200px;

background-color:#CC99CC;

border:solid 1px #CC99CC;

}

/\* child DIV tags \*/

#huey {

position:absolute;

left:10px;

top:10px;

}

#dewey {

position:absolute;

left:90px;

top:20px;

}

#louie {

position:absolute;

left:50px;

top:90px;

width:200px;

}

</style>

</head>

<body>

<div id="donaldDuck">

<div id="huey">

<img src="capitalA.gif" width="54" height="54" />

</div>

<div id="dewey">

<img src="capitalB.gif" width="54" height="54" />

</div>

<div id="louie">

<p>Here is some text. And some more text. And so on and so on and so on.</p>

</div>

</div>

</body>

</html>

Here is the above [example](http://fog.ccsf.edu/~srubin/divchilde3.html) displayed.

Within a parent DIV box in the above example, I have positioned three child elements: two images and a paragraph of text.

This feature is extremely useful for creating a complex element, such as a nav bar, which has many component pieces which need to retain consistent positioning in relation to one another, but which, as a whole, you might want to position at different locations on different pages. If you didn't have this parent/child positioning, you would have to recalculate the positions of ALL of the various elements when you moved your navbar; this way, you only have to move the parent element, and all of the child elements follow suit.

When positioning parent/child elements, there are only a few rules to keep in mind:

1. The parent DIV box MUST be LARGER than the child elements positioned within it. In other words, all of the child elements MUST fit COMPLETELY within the parent DIV box. Again, you must do this in order to ensure cross-browser compatibility; the browsers don't handle overflow of content in a consistent manner.

2. You can NOT nest DIV tags more than ONE level deep! In other words, you can't have a child positioned element within a child element; this is because the browsers handle "tertiary" nestings in a mutually incompatible and inconsistent manner.

As you can see, the restrictions on this sort of CSS-P can make it somewhat difficult to use for general purpose web page layout. We are rapidly moving into more advanced concerns as we brush the edge of DHTML. In DHTML work, this sort of construction is used very frequently, especially for nav bars and application interface elements.