

CSS Page Layout

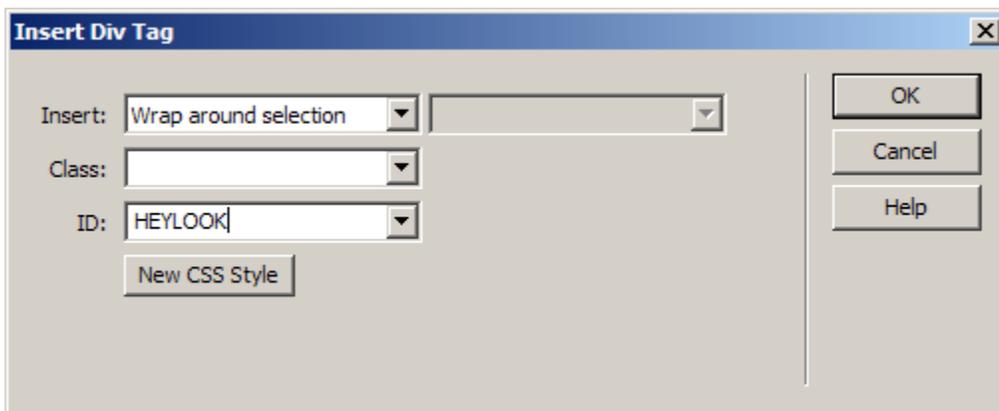
As we move beyond tables for designing & organizing our pages we meet the DIV tag. The <div> tag defines a division/section in a document. They begin life in a “design neutral” way with no particular attributes of any kind. There are two branching approaches to using divs for CSS layout.

1. Wrapping Your Existing Elements

If you already have a web page before you with text and perhaps graphics start by selecting some area of the page you’d like to format. For example select a paragraph and then follow these steps.

- Insert
- Layout Objects
- Div Tag

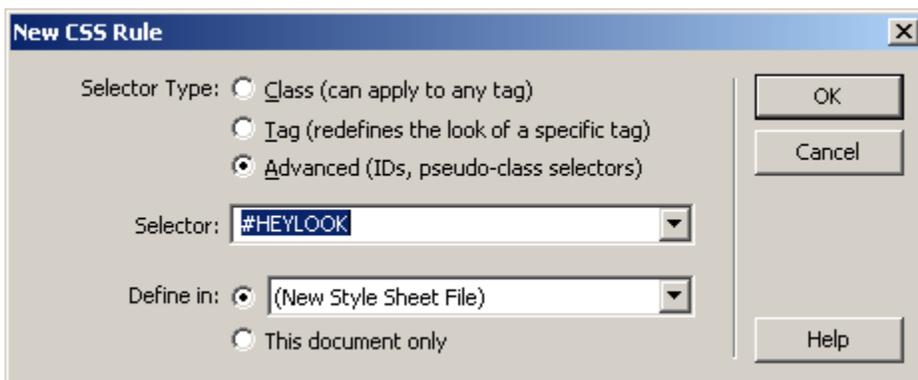
This dialogue box will appear.



Notice the words “Wrap around selection” This is what we want this time.

Also notice the ID: box – I’ve typed the name HEYLOOK in this spot. IT IS A **CRITICAL STEP** TO GIVE YOUR DIV TAG AN ID. It’s important because it is this name that we will be using to format as a CSS style. The name should be a sensible nickname for the area you are wrapping – for example something like “Links” or “Main” or “Header”. Click OK.

Now with the div area still selected (or reselected) open the CSS panel. And click the “New CSS rule” button. Something like this will appear.



Notice the "Advanced" choice is selected and the name of our DIV is the selector. The "Define in:" choices mean the same things they did in earlier classes about CSS:

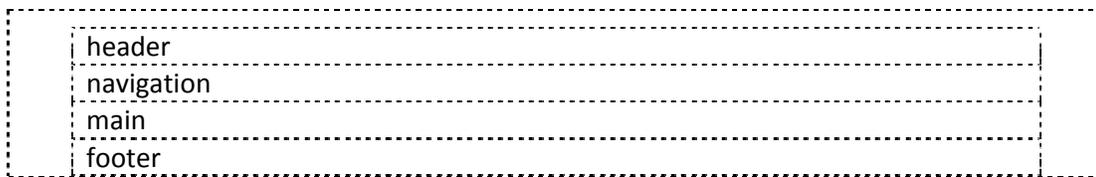
- New Style Sheet File means LINKED
- If linking a style-sheet to multiple documents use "new style sheet file"
- This Document Only means EMBEDDED
- If your ultimate goal here is building a template page, "This document only" is fine.

At this point you begin setting up your style for this DIV controlling issues such as font, background, margins and position. As you make these choices, if you hit the APPLY button you should see your DIV expressing the choices you've made.

2. Create Spaces for Content

The other branch of this design road is to start with no content present on the page. Merely insert as many DIVs as you want to hold the differing elements you will place on the page. Carefully make sure you give them IDs (remember how important that is?) then once again select them and create styles for each. Later on, add whatever content you want to these areas by typing text or dragging it in, and pasting or inserting graphics.

For example:



(wrapper)

These are div tags. The outer containing div we will call something like "wrapper" for example. When we inserted it we simply did so at the top of the page, or "at insertion point". We gave it the ID wrapper, hit ok and immediately went to the CSS panel to click: "New CSS Rule".

Our div will come up listed as an ID or top level div and the name will look like this #wrapper. This div is going to play the very important role of holding and positioning the rest of our divs inside it...as "sub" divs or "Compounds" the direct holders of our diverse content. Whatever attributes we give to the top level or containing div, flow down and are inherited by the sub divs. We can change this and give them different attributes if we want them to diverge from the rules of their "Parent" div.

If I make the wrapper div take up 90% of the screen width and sit in a centered position, all the divs I add inside it will follow the inherited rule. When adding the sub or compound divs we have to stop and consider our choices for where they should be entered. When we added **wrapper** we simply chose "at insertion point" but when adding header we might choose "After start of tag -> **wrapper** " top put inside, near the beginning. Then when adding "navigation" we would choose "After tag -> **header** " to make sure that this one landed cleanly AFTER the header div. It takes some getting used to, but it's very logical.