

3-COLUMN CSS LAYOUT

CSS and Div

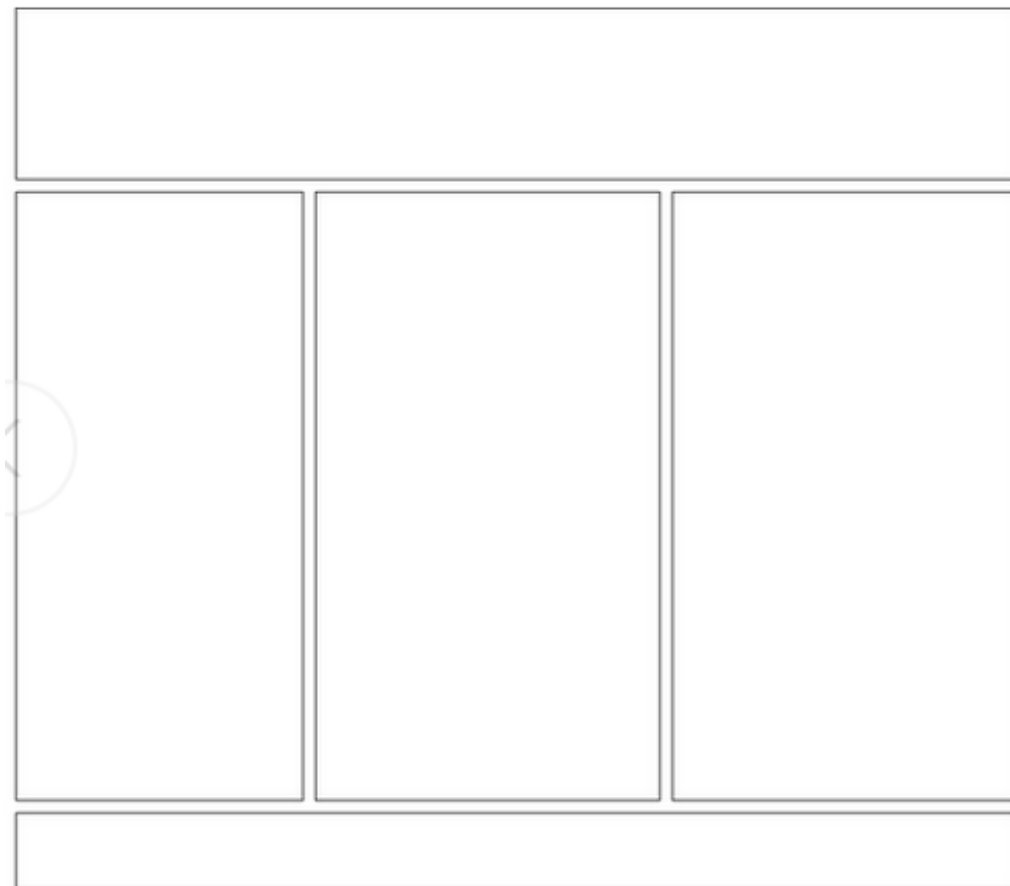
The div tag is used to specify a section within an HTML document. Anything from text to images to videos can be placed within a div. Divs are similar to tables but they are easier to use, customizable with CSS, and load faster than tables. Divs also make it easier for search engine spiders to “crawl” or examine a website – making your website easier to find via Google.

Div ID vs. Div Class

Div id should be used to identify a single element that appears only once within a document. Div class should be used to identify elements that appear more than once within a document. For example, let's say that you have an HTML document that uses a div for the header and a div to contain each image. a div id will be used to represent the header while a div class will represent the images

How to Layout your website

Step 1: Draw your Layout



You can draw your layout on paper or in a graphics program. If you've already got a wire-frame or even more extensive design in mind, you should simplify it down to the basic boxes that make up the site. This design has 3 columns in the main content area, as well as a header and footer.

Once you have your layout drawn out, you can start thinking of dimensions.

Step 2: The HTML structure

The HTML outline you need for this 3-column layout looks like this:

```
<div id="body">
  <div id="header"></div>
  <div id="main">
    <div id="content-1"></div>
    <div id="content-2">
      <div id="content-2-1"></div>
      <div id="content-2-2"></div>
    </div>
  </div>
  <div id="footer"></div>
</div>
```

Here's what those elements are used for:

- **#body:** A container element used to control the overall width of the layout and to center it horizontally
- **#header:** A place to put your normal header stuff. It's not actually part of the 3-column layout, but included here to make the layout a "complete" page.
- **#main:** The element that contains the content that will be displayed in columns.
- **#content-1:** The element that will become the left column. This will often be a vertical submenu.
- **#content-2:** A container element for the middle and right columns.
- **#content-2-1:** The element that will be the middle column. The main content normally goes here.
- **#content-2-2:** This element will be the right column, often used as a "sidebar".
- **#footer:** Your footer stuff goes here. Like #header, this is not part of the columns but included for the sake of completeness.

You could argue that the #main and #content-2 wrapper elements are unsemantic and unnecessary, and you would be correct for the most part. However, while there are ways of creating a 3-column layout without the #main and #content-2 wrapper elements, in my experience using these elements makes the layout more robust and the CSS easier to understand.

The wrapper elements are also very useful when you're applying design to the layout. If you leave them out you will often end up having to add them anyway in order to achieve the visual design you want. The only downside to using them is that they will make the HTML a few bytes larger and that you won't be able to (easily) rearrange the layout completely without changing the HTML.

With that bit of background out of the way, let's move on to the CSS.

The unstyled HTML

Start by taking a look at how your browser renders the unstyled document with a bit of placeholder content added in 3-column CSS layout.

Step 3: Overall width and some colors

Now let's add a bit of basic CSS to make it easier to see what we're doing. The following makes the entire layout 960px wide and horizontally centered, and gives most of the elements a background color:

```
html,
body {
    margin:0;
    padding:0;
    color:#000;
    background:#fff;
}
#body {
    width:960px;
    margin:0 auto;
    background:#ddd;
}
#header {
    background:#fdd;
}
#content-1 {
    background:#bfb;
}
#content-2-1 {
    background:#ddf;
}
#content-2-2 {
    background:#dff;
}
#footer {
    background:#ff9;
}
```

When viewed in a browser, all elements are stacked vertically in a single column.

Step 4: Create the first two columns

To start creating the columns, we'll give #content-1 and #content-2 widths and float them in opposite directions:

```
#content-1 {  
    float:left;  
    width:240px;  
    background:#bfb;  
}  
#content-2 {  
    float:right;  
    width:720px;  
}
```

The #content-1 element now makes up a column on the left side of the page, while #content-2-1 and #content-2-2 are stacked vertically on the right side of the page.

The #footer element looks weird at this stage, but that will be fixed later.

Step 5: Create the nested columns

To turn the #content-2-1 and #content-2-2 elements into columns, all we need to do is repeat what we did in Step 4 – give the elements widths and float them in opposite directions:

```
#content-2-1 {  
    float:left;  
    width:480px;  
    background:#ddf;  
}  
#content-2-2 {  
    float:right;  
    width:240px;  
    background:#dff;  
}
```

Step 5 now has the three columns we wanted, with #content-1 to the left, #content-2-1 in the middle, and #content-2-2 to the right.

Step 5: Push the footer down

The #footer element is currently displayed below #content-1, which is not where we want it to be. There are different ways of pushing it down below the three columns, but the most straightforward way is to use clear:both:

```
#footer {  
    clear:both;  
    background:#ff9;  
}
```

Step 6: Clean things up a bit

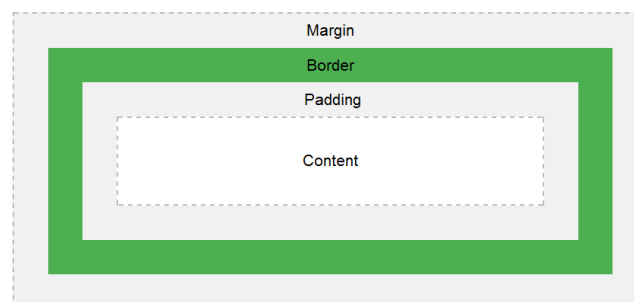
We're pretty much done at this stage, but to make the final demo page look just a little bit less ugly, let's give the main containers some padding:

```
#header {  
    padding:10px;  
    background:#fdd;  
}  
#content-1 {  
    float:left;  
    width:220px;  
    padding:10px;  
    background:#bfb;  
}  
#content-2-1 {  
    float:left;  
    width:460px;  
    padding:10px;  
    background:#ddf;  
}  
#content-2-2 {  
    float:right;  
    width:220px;  
    padding:10px;  
    background:#dff;  
}  
#footer {  
    clear:both;  
    padding:10px;  
    background:#ff9;  
}
```

Note that horizontal padding increases an element's total width, so to keep the same widths we need to subtract the horizontal padding from the width for each element – #content-1 becomes 220px wide instead of 240px since it has a total horizontal padding of 20px.

(According to the Box model, **Total element width** = width + left padding + right padding + left border + right border + left margin + right margin while

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin)



As a final touch we'll also add a class name of cf to some of the elements. This applies a couple of rules based on a technique that will clear floats without structural markup. Doing this makes the layout more robust since any floated elements added later on will be contained within the elements that these rules are applied to, and it will fix a couple of visual glitches in IE 6 and IE 7.

The elements that we apply the cf class to are #header, #main, and #footer. This will eliminate the need for clear:both on the #footer element, so we'll remove that as well.

```
.cf:after {  
    display:block;  
    clear:both;  
    height:0;  
    visibility:hidden;  
    content:" ";  
    font-size:0;  
}
```

Complete Code

HTML

```
<html>  
<head><title>Three Column Layout</title>  
<link rel="stylesheet" href="styles.css">  
</head>  
<body>  
    <div id="body">  
        <div id="header" class="cf">  
            <h1>3-column CSS layout, Step 6</h1>  
<p>Check <a href="http://miuc.ac.ke/">Our Schools Site</a> for info  
on what this is.</p>  
  
<p>This is <code>&lt;div id="header"&gt;</code>.</p>  
  
</div>  
<div id="main" class="cf">  
    <div id="content-1">  
<p>This is <code>&lt;div id="content-1"&gt;</code>.</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do  
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad  
minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip  
ex ea commodo consequat.</p>  
</div>  
<div id="content-2">  
    <div id="content-2-1">  
<p>This is <code>&lt;div id="content-2-1"&gt;</code>.</p>  
<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do  
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
```

minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

<p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

<p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

</div>

<div id="content-2-2">

<p>This is <code><div id="content-2-2"></code>.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>

<p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>

</div>

</div>

</div>

<div id="footer" class="cf">

<p>This is <code><div id="footer"></code>.</p>

</div>

</div>

</body>

</html>

CSS

```
html,body {
    margin:0;
    padding:0;
    color:#000;
    background:#fff;
}
#body {
    width:960px;
    margin:0 auto;
```

```

        background:#ddd;
    }
    #header {
        padding:10px;
        background:#fdd;
    }
    #content-1 {
        float:left;
        width:220px;
        padding:10px;
        background:#bfb;
    }
    #content-2 {
        float:right;
        width:720px;
    }
    #content-2-1 {
        float:left;
        width:460px;
        padding:10px;
        background:#ddf;
    }
    #content-2-2 {
        float:right;
        width:220px;
        padding:10px;
        background:#dff;
    }
    #footer {
        padding:10px;
        background:#ff9;
    }
    /* Easy clearing of floats (see https://www.sitepoint.com/simple-
clearing-of-floats/) */
    .cf:after {
        display:block;
        clear:both;
        height:0;
        visibility:hidden;
        content:" ";
        font-size:0;
    }

```