

LAB 03 – 3 COLUMN LAYOUTS WITH CSS

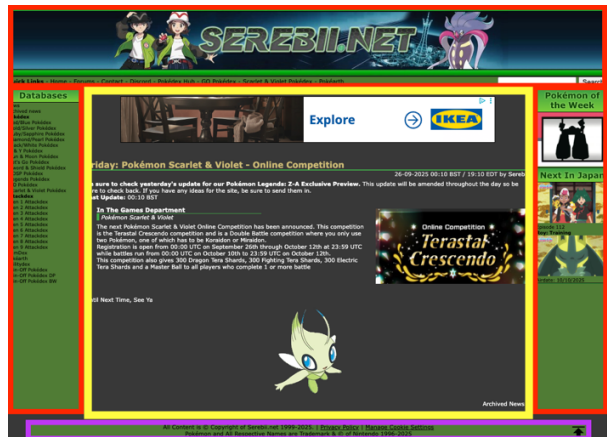
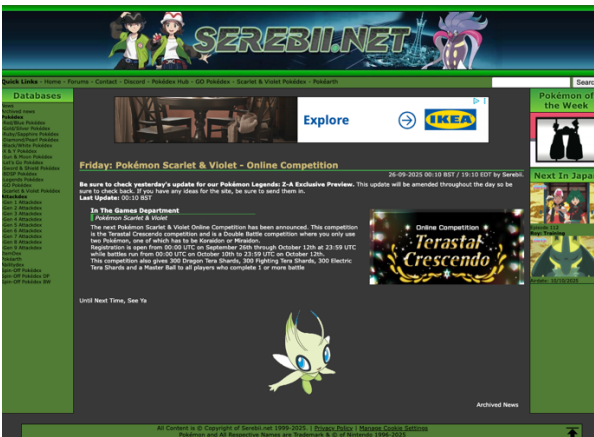
READINGS:

- ⇒ Positioning with CSS
- ⇒ CSS Float Property

WE WILL UTILIZE CSS TO CREATE A TABLE-LIKE FORMAT

In this exercise, we will create a web page that uses a three-column layout. It will include a header and footer that do not rely on table tags for formatting. This is a common technique for many simple web pages.
(no unnecessary complexity!)

(credit: serebii.net)



THIS PAGE IS DIVIDED INTO 5 BASIC SECTIONS:

1. 1 Header region at the top
2. 1 Footer region at the bottom
3. 1 Left column
4. 1 Right column
5. 1 Center column

LET'S TRY IT!

⇒ Create the CSS stylesheet

Start by defining the styles rules for our three columns in a file called `threecol.css` in your web site directory:

```
#left {
    float: left;
    width: 210px;
    border: dashed blue
}

#right {
    float: right;
    width: 200px;
    border: dashed blue
}

#center {
    margin-left: 210px; /* 218px works better, why? */
    margin-right: 210px;
    border: solid blue
}
```

⇒ Create the HTML page

Now, let's create a page called "lab3.html" in your web site's directory that uses the `threecol.css` style sheet. Place the three columns onto your HTML page like this:

```
<div id="left">
    <!--left content -->
    Left
</div>
<div id="right">
    <!--right content -->
    Right
</div>
<div id="center">
    <!--center content -->
    Center
</div>
```

Combining this HTML and CSS should yield a page that looks something like this:



NOTE:

Notice that 1 div *floats* to the left & takes up 210px, while the other *floats* to the right. The third is in the middle, taking up the remainder of the space. It is generally best practice to use relative sizes (20%, etc.) than pixels.

⇒ Add the header

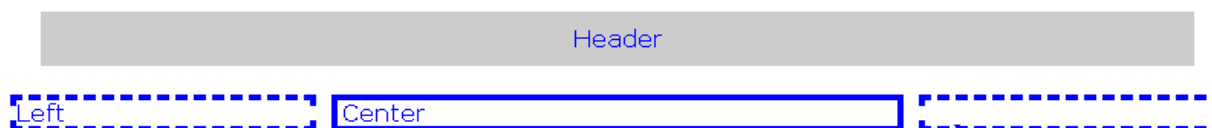
Now we can add the top header style in our CSS with the following rules:

```
#top {
    margin: 20px;
    padding: 10px;
    text-align: center;
    background: #ccc;
}
```

& add another div to our HTML:

```
<div id="top">
    <!--header content -->
    Header
</div>
```

Now you should have this page:



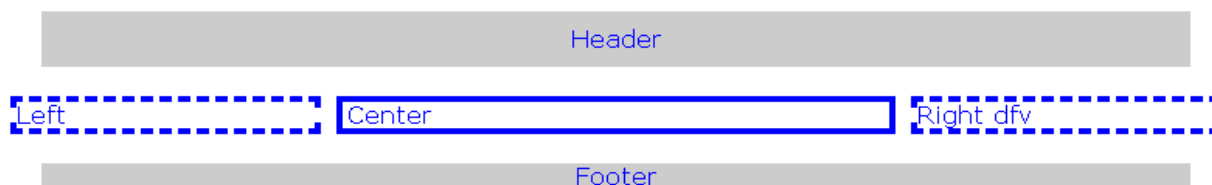
Let's add a footer to the stage. We'll start with the style:

```
#footer {
    background: #ccc;
    margin: 20px;
    text-align: center;
}
```

& for the HTML:

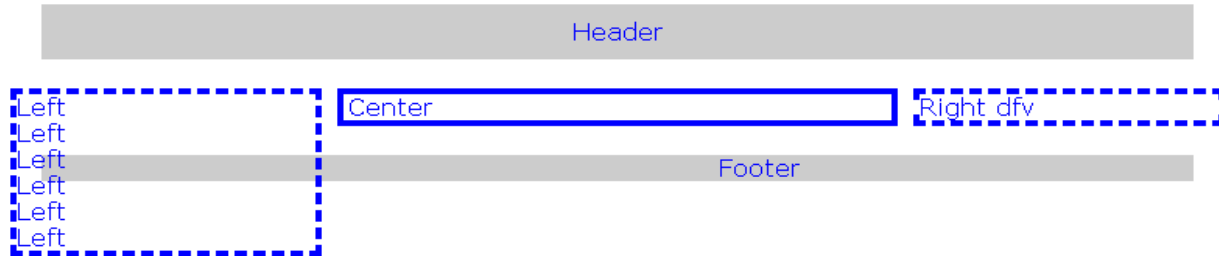
```
<div id="footer">
    <!--footer -->
    Footer
</div>
```

This gives us this page:



The problem with this page is that the left and right columns float *on top* of the page. The position of the footer is determined solely by the height of the center column.

Let's see what happens if you add content to the left div like this:



That's not really what we wanted, is it? One trick is to simply modify the footer style so that it looks like this:

```
#footer {
    background: #ccc;
    margin: 20px;
    text-align: center;
    clear: both;
}
```

- ⇒ This forces the footer to appear below anything floating to the left or the right
- ⇒ This clears the left and right of the footer div
- ⇒ For cases where the DIV contents overflow, investigate the *clearfix hack*
 - a. See the W3Schools Float reading above for more information

LAB REQUIREMENTS: REDO YOUR HOMEPAGE TO AVOID TABLES FOR LAYOUT

Use all the tools we have discussed above to create a CSS-focused version of your home page. Later, we will use the bootstrap framework to style in tabular format. For now, remove the tables and refactor the layout using DIV tags and CSS rules in your own stylesheet.

(END OF LAB 03)