Agenda

In this tutorial we are going to create a 3 column layout.
We will be structuring the elements and creating the neccessary css to position them.
While working with CSS we will encounter few issues and solve them on the way to make the design.

Look at the files.
Inside the folder we have
1)External CSS
2)Separate html for the columns
3)Main html file

Step1

Insert the content of sidebar1.txt file inside a <aside class = “sidebar1”> </aside
Copy the contents from sidebar1.txt and put them inside it.
Repeat the same steps but this time create an article tag and keep the contents of main.txt file inside the article.

Time to play with the css

Add the class

```
.sidebar1 {
float: left;
width: 20%;
}
```

- This class style floats the sidebar div to the left of the page, and gives it a width of 20%.
- The width property is important in this style: Unless you’re floating an image that has a set width, you should always set a width for a floated element.
- If not then the browser sets the width based on the content inside the float, leading to inconsistent results.

Results: Text in the main column reaches the bottom of the sidebar, it wraps around the bottom of the Sidebar. To make the main body text appear like a column of its own,we will add enough left margin to indent the main text beyond the right edge of the sidebar

Step2:

Create a style for the second column:

```
.main {
margin-left: 22%;
}
```

Since the sidebar is 20 percent wide, a margin of 22 percent indents the main content an additional 2 percent, creating a spacing between the two columns. It looks better with spacing

Step3

Open the file sidebar2.txt . Copy all the HTML from that file, and then return to the start.html file.
The HTML for this next column goes after the main text inside the <article> tag.
The tag is <aside class="sidebar2">

Create styling for it

```
.sidebar2 {
float: right;
width: 20%;
}
```

Save and see the Result!
The second sidebar appears below

step4

the main content and even runs over the footer.
. When We float an element, only HTML that comes after
that element wraps around and appears next to the floated element.Means since the second sidebar's HTML is after the main content, it doesn't
appear next to the main content, only after it.
Solution: simply float the main column as well. If you set its width
so all three columns add up to no more than 100 percent, then the three columns will sit side by side. For this example, you'll try that approach.
Edit the main style to

```
.main {
float: left;
width: 60%;
}
```

Now see the results!

everything's turned black! It is the background color of footer that is spread over here
Why?when we float an element and another element wraps around it, the background color and borders for that element actually extend underneaththe floated element.
Fix Add one more style after the .sidebar2 style:

```
footer {
        clear: both;
}
```

Lets Play with the Padding

Add padding to the .sidebar1, .main, and .sidebar2 styles, so the styles look like this:

```
.sidebar1 {
float: left;
width: 20%;
padding: 0 20px 0 10px;
}
.main {
float: left;
width: 60%;
padding: 0 20px 0 20px;
}
.sidebar2 {
float: right;
width: 20%;
padding: 0 10px 0 20px;
}
```

Save and see the Results!

What Happened?

By adding padding, you've
essentially increased the width of each column, and since the total widths of the columns (20% + 60% + 20%) already totaled 100%, that extra padding forces the third column down below the other two.

Solution1:
remove the padding from the styles in the CSS file and then,
in the start.html file, add a inside each of the columns, something like this:

```
<aside class="sidebar1">
<div class="innerColumn">
... content goes here ...
</div>
</aside>
```

Then, in the styles.css file, simply create a style to add the padding:

```
.innerColumn {
padding: 0 20px 0 10px;
}
```

Problem with this approach is you need to add additional HTML

Better solution

Add another style to the style sheet:
* {
-moz-box-sizing: border-box;
box-sizing: border-box;
}}
This style takes advantage of the universal selector

It applies the
box-sizing property to every element on the page.

By setting this property to
border-box,We are instructing web browsers to use the size of the padding and border properties as part of the CSS width value. Means, the padding doesn’t add to the CSS widths

Only problem is that **Internet Explorer 7** and earlier don’t
understand the box-sizing property. We can ignore it. We shouldnt go below IE 7 in general

Beautify

adding a left and right border

```
.main {
float: left;
width: 60%;
padding: 0 20px;
border-left: dashed 1px rgb(153,153,153);
border-right: dashed 1px rgb(153,153,153);
}
```

Play with width

the page is a liquid design, meaning that it expands to fill the entire width
of the browser window. Lets restrict to the max lenght. Minimu m for now is acceptable

add a new tag:
<body>

And add the styles

.pageWrapper {
max-width: 1200px;
margin: 0 auto;
}

The max-width property provides a liquid layout but only up to a point. In this case, when the browser window is
wider than 1200 pixels, the div won’t get any wider.
The margin setting here—0 auto—provides 0 margin for the top and bottom, and auto margins for left and
right. That auto setting tells the browser to automatically figure out the margin, splitting the space evenly between left and right and centering the div in the browser window.

Thanks

Slides By--

Sumit Kumar