

More CSS

1. Colors

You can add basic colors just with their names, like

```
color: red;
```

But if you want to have a special color you need to use RGB-coding. RGB means you mix the color by yourself like if you had water colors. R = Red, G = Green, B = Blue. The amount of each color is given in hexadecimal values from 0 to 15. The values are written with numbers from 0-9 and the from 10 to 15 with alphabets a to f. So 0 means no color and f is the maximum amount.

Examples:

```
color: #36d;    Means the amounts of the color mixed are Red=3, Green=6, Blue=d (means 13).
```

```
color: #36eedf; This is a way to give more strict values of each color. Now Red=36, Green=ee and Blue=df.
```

To see the codes of colors you can use for example Gimp. Open the color palette and you will see the color and its RGB value. There are also color maps in the internet.

2. Background image

```
background-image:url("imagenam");
```

As default the image will repeat until it fills the whole element. That can make a nice wallpaper, but if you don't want it add

```
background-repeat: no-repeat;
```

Or

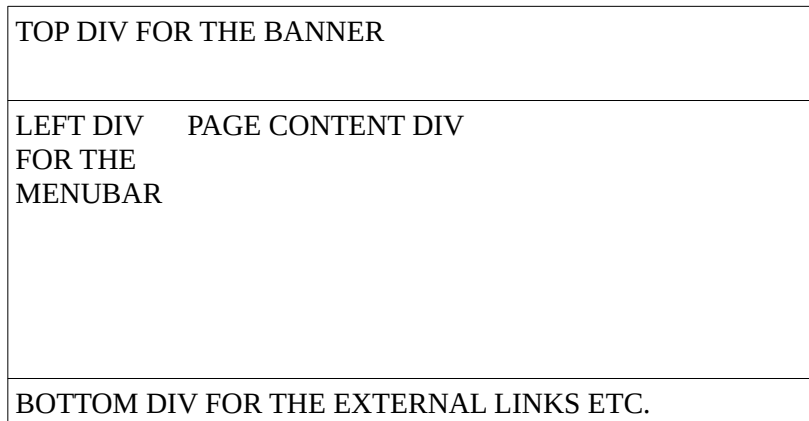
```
background-repeat: repeat-y;    or background-repeat: repeat-x;
```

Try it!

3. Making the layout

To make the layout you have to use div -elements. With divs you can make a grid where every element is located on the place you wanted. Usually a page layout has a top div for a banner, some content divs and a footer div for some usefull links, copyright text and so on. All the content divs should be wrapped in to a one big div because it makes easier to move and adjust the whole page

So the layout could look for eaxmple like this:



And all this should be inside one big div. As you see, all the divs are different and need a different CSS definitions. That is achieved by giving different identities to all divs with attribute id. It works in the same way as the class attribute but there is one difference: you can use class on the page several times, but id is just for a certain element, like there can be many pupils in a class but they each have their ids and none can have the same.

Example html:

```
<div id="wrapper"> <!-- WRAPPER DIV , AROUND ALL OTHER DIVS-->
<div id="top">
TOP BANNER
</div>
<div id="left">
LEFT DIV FOR THE MENUBAR
</div>
<div id="middle">
MAIN CONTENT HERE
</div>
<div id="bottom">
BOTTOM
</div>
</div> <!-- CLOSING THE WRAPPER DIV-->
```

Important notifications, read first!

In CSS you have to define what the divs look like and how are they located. Define the widths and heights for every div in percentages if you want to the page to scale on every screen; computers, tablets, phones etc. Remember also, that the highest value for the height and width is 100%, so if the left div's width is 25%, the other divs aside can be no more than 75%!

If you add padding to a div, it will enlarge as much as the padding takes room. So a div width 25% and padding-left:5% would be summed as 30%. To avoid that place everything in the divs inside other elements, for example a text inside `<p>` elements and add the padding for them.

Example: `<div> <p> text here </p> </div>`

Now you could add padding for `<p>` without affecting div's measurements and ruining possibly the layout.

Floating side by side

To get the elements side by side you can use floating. Add to CSS `float:left;` for all the elements to see it happen. If it doesn't work check that there is enough room for it. If the width of the elements together is over 100% they cannot be side by side.

Example CSS for the previous html:

```
html, body
```

```
{  
height:100%;  
}
```

```
// That definition was just a trick to make the height working in the divs.
```

```
div#wrapper
```

```
{  
width:60%;  
height:98%;  
border:2px groove #33e;  
}
```

```
div#top
```

```
{  
width:100%;  
height:25%;  
background-color:#e33;  
}
```

```
div#left
{
height:70%;
width:28%;
background-color:#ee3;
float:left;
}
```

```
div#middle
{
height:70%;
width:72%;
background-color:#ddd;
float:left;
}
```

```
div#bottom
{
width:100%;
height:5%;
background-color:#e33;
}
```

Save and look the result in the browser. It's not working as it should; bottom element does not have the background color. The bottom element does not take it's room unless you clear the the previous elements, add to the CSS of the bottom:

```
clear:both;
```

That will do the job and your layout is now ready. Now you can add some content inside the divs, but remember to put it inside other elements, for example text inside `<p>` and `</p>`, then you can add padding for the p -element in the CSS.

Finally, to center the whole thing automatically, add for the body CSS definion `text-align:center;` and for the wrapper definitions `margin-left:auto; margin-right:auto; text-align:left;`