Page banner, sidebar, main content, footer

Old method: Use `<table>`, `<tr>`, `<td>`

New method: `<div>` and "float" CSS property

Think of your web pages like water. With "float" your `<div>`'s will automatically move around to allow other `<div>`'s to fit.
## Common layout

<table>
<thead>
<tr>
<th>Banner (Header)</th>
<th>Navigation (Menu)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sidebar (News, etc)</th>
<th>Main page content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Footer (Copyright, Contact information) |
The infamous "box model"
<table>
<thead>
<tr>
<th>Banner (Header)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Navigation (Menu)</td>
</tr>
<tr>
<td>Sidebar (News, etc)</td>
</tr>
<tr>
<td>Footer (Copyright, Contact information)</td>
</tr>
<tr>
<td>Container (wrapper)</td>
</tr>
</tbody>
</table>
<div id="wrapper">
  <div id="header"></div>
  <div id="menus"></div>
  <div id="sidebar"></div>
  <div id="main"></div>
  <div id="footer"></div>
</div>
<div id="wrapper">
	<div id="header">My portfolio</div>
	<div id="menus">Home | Contact | About</div>
	<div id="sidebar">Today's news</div>
	<div id="main">Here are my photos...</div>
	<div id="footer">Copyright 2009, Me</div>
</div>
<table>
<thead>
<tr>
<th>My portfolio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
</tr>
<tr>
<td>Today's News</td>
</tr>
<tr>
<td>Here are my photos...</td>
</tr>
</tbody>
</table>

Copyright 2009, Me
CSS style for each id

#wrapper {...}
#header {...}
#menu {...}
#sidebar {...}
#main {...}
#footer {...}
width: and float:

```css
#wrapper {width: 750px;
    margin: 0px auto;}
#header   {...}
#menu     {...}
#sidebar  {width: 200px;
    float:   left;}
#main      {width: 550px;
    float:   right;}
#footer     {...}
```
With CSS positioning added

My portfolio

Home | Contact | About

Today's News

Here are my photos...

Copyright 2009, Me
Tables are an easy way to break material into columns and rows.

Tables should be used for information in a spreadsheet form. Columns and rows should be related. Use tables to look up information.

Layout with <div> when columns and rows are not related to each other.

Examples:
- Team schedules are tables: dates, locations, and scores are related to each other.
- News page with menu and ads are not tables: Ads do not relate to news content, and menus appear on all pages.
The sidebar often appears on the right instead of the left.

Navigation menus often appear in more than one location on the page.

Main content may be split into two columns.

Simpler is better. Too many columns and sections becomes confusing, and also hinders accessibility for blind users or others using by non-standard equipment.
The infamous "box model"

The box model is described in full at:
http://www.w3.org/TR/CSS2/box.html

Not easy to understand, but important.

Remember nesting, from the first class?  <div>'s can nest inside other <div>'s. Each <div> has width, margins, borders, padding and content. Content is the "real" part.

Often, margins, borders and padding will be zero, or set automatically. Together with the "float" they will position your content.

More explanation at:
Chapter 10 in your textbook
http://redmelon.net/tstme/box_model/
http://www.brainjar.com/css/positioning/default.asp
First, create a "wrapper" `<div>` . Specify the width, to confine all other `<div>`'s inside it.

Then, nest all other `<div>`'s inside it.
In this example, we do not have content yet. We will place content between the header `<div></div>`, the menus `<div></div>`, etc.

All tags can have an **id** attribute.

Each tag that uses an id attribute should use a different value for the id. No duplicate id's on the page. (You can use the same id on different pages. So, if you have two files, and both have a tag that has **id="content"** that is okay.)

The `<div>` with the id "**wrapper"" surrounds (encloses) all the other `<div>`'s on the page. The other `<div>`'s are "nested" inside.
<div id="wrapper">
  <div id="header">My portfolio</div>
  <div id="menus">Home | Contact | About</div>
  <div id="sidebar">Today's news</div>
  <div id="main">Here are my photos...</div>
  <div id="footer">Copyright 2009, Me</div>
</div>

Now there is content between the tags, but no style.

<div> without style is like <p>.
Without the CSS position properties, the text will appear as paragraphs in the same order as the source code:

header, menus, sidebar, main and footer will appear one below the other on the page.

Set widths, margins, etc. and add position properties to fit the pieces together like Lego blocks.
CSS sections start with a "selector". We have used tag names as selectors:

```
p {font-size: 32px;
     color: #ff00c3;}
h2 {background-color: #ffff00;}
```

We have used "class names" as selectors. Class names start with a period in the CSS:

```
.hello {color: red;}
```

Now we will use id names as selectors. Id selectors start with a #

```
#main {font-size: 24px;
      font-weight: bold;}
```
The wrapper surrounds all the other <div>'s. It is set to 750px wide. **ALL <div>'s nested inside must fit inside 750px.**

<div id="sidebar">..."/div> will be 200px; and will appear on the left. <div id="main">..."/div> will be 550px, and will "float" up and to the right to fill in the space next to <div id="sidebar">..."/div>. Together, they will add up to 750px – the maximum width that can fit inside the <div id="wrapper">..."/div>.

Try working through the steps in the lesson **Simple 2 column CSS layout** Click that link in the list at: http://research.gallaudet.edu/ART290/
With CSS it will eventually look like this example.