

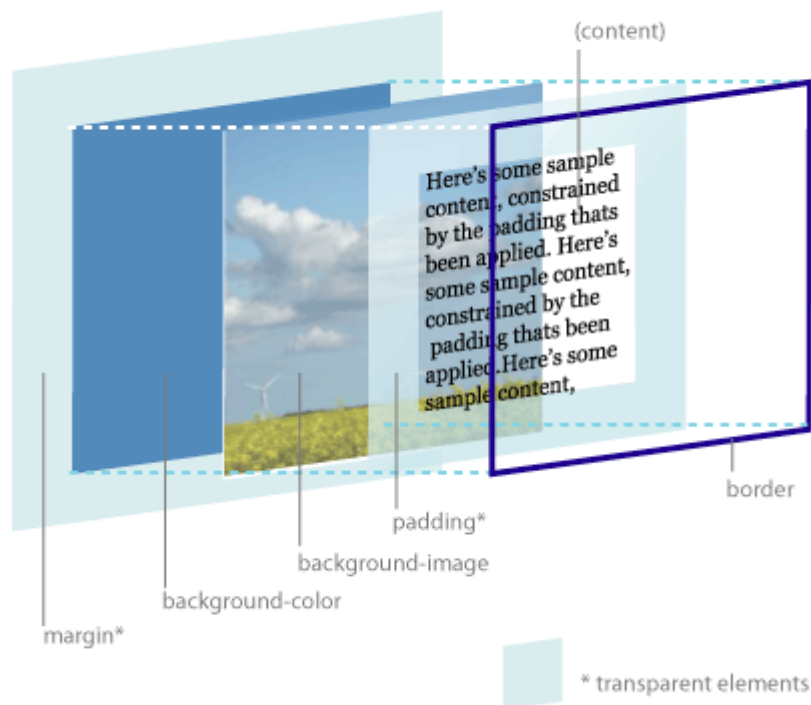
lab1.bboxes

example boxes/

This lesson will cover the fundamentals of the CSS Box Model, examining the height, width, padding, margin and border properties of a CSS box, and a brief overview of alternative positioning using the margin property.

The CSS Box Model

THE CSS BOX MODEL HIERARCHY



used with permission, created by Jon Hicks: <http://www.hicksdesign.co.uk/>

Basic Positioning

1. Viewing **#box1**
 - height:200px;
 - width:200px;
 - padding:3px;
 - margin:10px;
 - border:3px solid #f00;

2. Viewing **#box1a**
 - height:100%;
 - width:100%;
 - background:#ccc;
3. Absolute Positioning **#box1**
 - position:absolute;
 - top:200px;
 - left:200px;
4. Relative Positioning **#box1**
 - position:relative;
5. Viewing **#box2**
 - position:relative;
 - height:100px;
 - width:100px;
 - padding:3px;
 - margin:10px;
 - border:3px solid #0f0;
6. Absolute Positioning **#box1** (*again*)
 - position:absolute;
 - top:200px;
 - left:200px;
 - **Notice:** Watch how **#box2** is now relatively positioned above **#box1** – non-nested relatively positioned elements disregard surrounding absolutely positioned elements
7. Absolute Positioning **#box1a**
 - position:absolute;
 - top:20px;
 - left:20px;
 - **Notice:** The nested element, when absolute positioned is used on both elements, gets its X Y coordinates from the parent element, **not** from the browser window

Positioning w/ Margin

1. left: experiment **#box2**
 - position:relative;
 - left:100px;
 - delete the height, width and margin
 - **Notice:** The right 100px are missing
2. set only margin-left:
 - margin-left:100px;
3. or set all margins independently change margin-left: to margin:
 - a. margin:top right bottom left;
 - b. start with margin:100px 0 0 0;
 - c. then change to margin:100px 100px 0 0;
 - d. and finally margin:100px 100px 300px 200px;

lab2.floats

example floats/

This lesson will cover the fundamentals of floating CSS elements and how floats can be leveraged to create columns.

Basic Floating

1. setting the colors
 - color **#col1**
 - background:#ccc;
 - color **#col2**
 - background:#eee;
 - top border **#footer**
 - border-top:1px solid #00f;
2. float **#col2**
 - float:right;
 - width:200px;
3. Cut & Paste the **col2 div** above the **col1 div (index.htm)**
 - **Notice:** Watch where the floating content moves – to the right of the navigation
4. Move the **col2 div** back to below the **col1 div (index.htm)**
5. Remove **#col2** float & width
 - **Note:** If you float **#col1** left it will behave as expected and the content in the **col2 div** will “float” around it

Creating Simple Liquid Columns

1. setting the widths
 - **#col1** width:30%;
 - **#col2** width:70%;
2. floating the navigation
 - **#col1** float:left;
3. floating the content
 - **#col2** float:right;
 - **Notice:** The footer no longer clears col1
4. Clearing the **#footer**
 - clear:right;
5. Removing content from the **col1 div (index.htm)**
6. Clearing the **#footer**
 - clear:both;
7. undo the float right **#col1** float:left; **#footer** clear:left;

This lesson will apply the CSS box model and floating fundamentals to a simple site with a header, two columns and a footer.

Examining the Site

1. View the site without the style sheet
2. Examine the XHTML of **index.htm**
3. Attach the style sheet
 - Type “style.css” with double quotes into url();
4. Every element has its own “box” -- the box model is everywhere
5. Points of Interest
 - applying styles to the **body** tag
 - text-align:center;
 - the font size is set by a percent
 - use of **#wrapper**
 - contains all the surrounding elements
 - gives design added flexibility
 - *Example #wrapper:*
 - width:700px;
 - text-align:left;
 - hiding the **h1** text replacing with background image
 - The **h1** tag is a display:block by default, the height is set to a fixed pixel height and the overflow is hidden. When the padding top of the **.hide** class is set to a value greater than the fixed height of the **h1**, the content contained within is hidden. The background image of the **h1** is used for the page title.
 - **Hey, that whole h1 thing seems foolish.** As you saw when the styles weren't applied to the site, this allows for a graceful degradation of the site into older browsers and increases accessibility for those using screen-readers while maintaining the semantically correct h1 tag.
 - the tabs
 - were created by modifying the **ul**, **li** and **a** XHTML tags
 - pay close attention to the use of padding to pull off the effect
 - this example can be extended in many ways, one of which would be the inclusion of background images to simulate things such as rounded corners
 - see: <http://www.alistapart.com/articles/slidingdoors2/>

- the left navigation images
 - padding is added to the **a** tag and the background image coordinates begin before the padding and fills the void
 - since Win IE does not fully support *:hover*, the **a** tag is used, on modern browsers, *:hover* can be added to the **li**'s
- the **col** widths don't total to 100%
 - Applying margins and borders can have unintended effects on surrounding elements due to the way each individual browser interprets the box model.
 - see: <http://tantek.com/CSS/Examples/boxmodelhack.html>