CSS 2 Notes:

CSS-P (or Cascading Style Sheets Positioning, a part of the unevenly implemented CSS2 standards) allows the web

programmer to position HTML-formatted resources anywhere they like on a web page.

 replaces tables as the primary means for page layout in HTML.

 CSS-P provides programmers with a means for creating application-like screens from positioned "layers" which, in combination with JavaScript, can be used to create simple "dynamic" HTML content (or DHTML).

NOTE: Where to place it?

you must use CSS-P only within the "splash" region of your HTML page (within the first 300-400 pixels in height, and 500-700 pixels in width) so that the user won't have to scroll to see the CSS-P-based content

Only one type of HTML tag should normally be positioned: the DIV tag.

DIV tag was originally created to mark a section division within a larger HTML page

ANY type of HTML content may be placed WITHIN the DIV tag which we're positioning.

 the positioned DIV tag will ONLY have CSS-P properties assigned to it; you won't be assigning regular text-formatting CSS properties to a DIV tag.

Text-formatting properties will only be assigned to the P tag, the heading tags, etc, as usual.

In the future, you will be able to cut out the middleman and assign CSS-P properties directly to ANY HTML tag, positioning that tag.

The DIV tag requires one attribute: ID. This ID will be set to a unique identifying name, and assigned appropriate CSS-P properties in EITHER a linked or embedded style sheet. Note: To ensure cross-browser compatibility, you MUST use an ID to assign CSS-P properties to a DIV tag.

<div id="fred">

<img src="capitalA.gif" width="54" height="54" />

</div>

Positioning:

elements are positioned from THEIR upper-left-hand corner in relation to the upper-left-hand corner of the web page.

#fred {

position:absolute;

left:100px;

top:30px;

}

Example (in context using an embedded style sheet):

<html>

<head>

<title>Example CSS-P Page</title>

<style type="text/css">

#fred {

position:absolute;

left:100px;

top:30px;

}

</style>

</head>

<body>

<div id="fred">

<img src="capitalA.gif" width="54" height="54" />

</div>

</body>

</html>

Minimum of THREE ( 3 ) properties:

The left property represents the x-coordinate value, and the top property represents the y-coordinate value; these properties are ALWAYS set using pixel values.  The position property will always be set to absolute.

ALWAYS define left and top properties using integer pixel-based values.

Position:

Relative is unreliable

Fixed positioning "fixes" that element in the web browser window so that other elements on an HTML page scroll up and down while the "fixed" element remains stationary, stuck in place.

absolutely positioned element is removed from the regular flow of HTML code and positioned in relation to the upper-left-hand corner of the web page

How to get text to wrap in CSS P?

Use width property

Overlap / Overlapping

you need to use the z-index property to set the z-axis (forward and back) value in this flattened three-dimensional space; z-index tells the browser which element goes in front and which element goes behind.

Property: **z-index**  
Values: 0 (zero) or any integer; (negative values are permitted).  
Example: **z-index:10;**

The higher the z-index value is, the farther in front an element becomes; the lower the z-index value is, the farther in back an element becomes. You may not use numbers with decimal places.

I have set fred's z-index to 10, and martha's z-index to 20; martha, therefore, is in front of fred.

If NO z-index value is set, or BOTH z-index values are identical, then whichever DIV element comes LATEST on the HTML page goes in front.

You can turn a positioned DIV tag into a colored box. This feature is particularly useful when creating application "screens" for DHTML purposes. Or

creating low-weight design elements without GIFs.

Width, Height, background-color

#fred {

position:absolute;

left:100px;

top:30px;

width:100px;

height:100px;

background-color:#CC99CC;

border:solid 1px #996699;

}

Last, but not least, you MUST set the border property for the positioned element.

The border MUST be the same color as the background color; this is due to flaws in CSS implementation in older Netscape browsers.

CHILD DIV TAGS

To produce consistent appearance and to control properly the distance between the edge of positioned content and the edge of a DIV box, one must put HTML content into positioned DIV tags which are NESTED INSIDE the positioned, colored DIV box.

<div id="parentDiv">

<div id="childDiv">

<p>Here is some text. And some more text. And so on and so on and so on.</p>

</div>

</div>

"parentDiv" which is 200x200

"childDiv" has been set to be 180 pixels wide.

 "childDiv" DIV tag's positioning is set at 10,10. Child nested elements are positioned in relation to the PARENT element, NOT to the web page as a whole. Therefore, "childDiv" is 10 pixels from the left and 10 pixels from the top of the PARENT element's upper-left-hand corner.

"childDiv" DIV tag has been set to be 180 pixels wide. Since it has been positioned 10 pixels from the left of the parent element, and the parent element is 200 pixels wide, I want to create an even "margin"

MAIN POINT:

because padding, border, and margin properties do not work consistently cross-browser in CSS-P, I have had to create a fake margin by positioning a child element absolutely within an absolutely positioned parent element.

When positioning parent/child elements, there are only a few rules to keep in mind:

1

The parent DIV box MUST be LARGER than the child elements positioned within it. In other words, all of the child elements MUST fit COMPLETELY within the parent DIV box. Again, you must do this in order to ensure cross-browser compatibility; the browsers don't handle overflow of content in a consistent manner.

2

You can NOT nest DIV tags more than ONE level deep!

NOTE

If you use the CSS3 elements of text-shadow and border-radius on a page then you should validate the page for CSS3.

The code for the validation icon:  
  
<a href="http://jigsaw.w3.org/css-validator/check/referer?profile=css3">  
<img style="border:0;width:88px;height:31px"  
src="http://jigsaw.w3.org/css-validator/images/vcss-blue"  
alt="Valid CSS!" />