Fixed and Fluid Layouts

Fixed layout:

- Pixel-based widths for whole page and for each column of content.
- Width doesn’t change on smaller devices or browsers.
- Commonly used on corporate and big-brand sites.
- Easiest when learning CSS.

Fluid (or liquid) layout:

- Uses percentages for widths.
- Page shrinks and expands depending on viewing conditions.
- Recently enhanced to create *responsive* and *adaptive* layouts.

Responsive, Adaptive, and Elastic Layouts

Responsive and adaptive layouts:

- Shrinks like traditional fluid layouts, but also shifts in specific ways based on screen size.
- Tailors to mobile, tablet, and desktop users using same HTML, not three different sites.

Elastic layout:

- Uses ems for both width and all other property sizes.
- Page scales according to a user's font-size settings.

Setting the Width for an Element

Max-width property: Ideal for setting outside limit of a fluid layout.

Min-width property: To prevent an element from getting too narrow, apply the min-width property.

```
#container {
background: url(../img/bg-bluebench.jpg)
repeat-y;
max-width: 950px;
width: 90%;
}
```

Making Elements Float

Make elements float in a sea of text (or other elements).

- Use this technique to make text wrap around images or figures to create multi-column layouts and more.
- When you float an element to a side, the content flows around it instead of after it.

Floating and Wrapping Text

FLOATED LEFT, SO TEXT WRAPS.

MARGINS SET TO 110PX, SO TEXT DOESN'T WRAP.

recent entries

Hospital Sant Pau

June 26, 2011

The Saint Paul Hospital at the top of Gaudí Avenue in the Sagrada Família neighborhood is an oft-overlooked gem of modernist architecture. Although the building was begun in 1902 under the direction of the architect Lluís Domènec i Montaner, the hospital itself dates from the 14th century. It serves some 34,000 inpatients yearly, along with more than 150,000 emergency room...

continued

Cathedral Cloister

June 24, 2011

Outside it feels like a battle is raging, with firecrackers going off left and right in honor of Sant Joan, but in the cloister of Barcelona's 12th century Cathedral, it's quiet enough to hear the trickle of the water from the fountain. As everywhere else in Barcelona, Saint George is slaying his dragon here. Somehow...

continued

recent entries

Hospital Sant Pau

June 26, 2011

The Saint Paul Hospital at the top of Gaudí Avenue in the Sagrada Família neighborhood is an oft-overlooked gem of modernist architecture. Although the building was begun in 1902 under the direction of the architect Lluís Domènec i Montaner, the hospital itself dates from the 14th century. It serves some 34,000 inpatients yearly, along with more than 150,000 emergency room...

continued

Cathedral Cloister

June 24, 2011

Outside it feels like a battle is raging, with firecrackers going off left and right in honor of Sant Joan, but in the cloister of Barcelona's 12th century Cathedral, it's quiet enough to hear the trickle of the water from the fountain. As everywhere else in Barcelona, Saint George is slaying his dragon here. Somehow...

continued

Main Div Floated Left, so Sidebar Flows Along Right Side

Clear Property

Market Day

June 23, 2011

Quiet returns. And the refrigerator was empty, so off I set on my way to the market. The plan was homemade sushi with friends so I made the trip down to the central Barcelona market: the Boquería.

I never get tired of coming here. There are rows and rows of busy stalls with fresh fruit and vegetables, and spices and nuts. And more than a couple of crazy tourists like me taking pictures. I confuse the locals by talking to them in Catalan, but...

continued

continued

archive

- May 2011
- Apr 2011
- Mar 2011
- Feb 2011
- Jan 2011

More from the archive

about this photoblog

This photoblog is the product of a love of computers, photography, and Barcelona. If you're interested in any of my photos, please contact me. The photographs on these pages are licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/2.5/; or, (b) send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

To Control Where Elements Float

Type **clear:**.

Type **left** to keep elements from floating to the left of the element you're styling.

Or type **right** to keep elements from floating to the right of the element you're styling.

Or type **both** to keep elements from floating to either side of the element you're styling.

Or type **none** to let elements flow to either side of the element you're styling.

Setting the Border

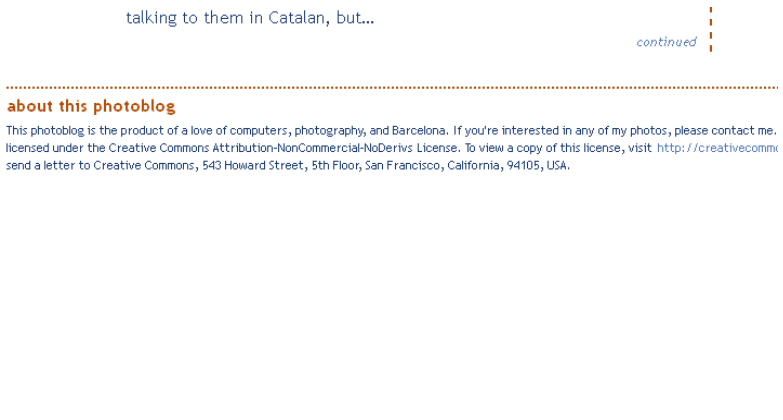
Create border around or on individual sides of an element and then set its thickness, style, and color.

If you've specified any padding, the border encloses both the padding and the contents of the element.

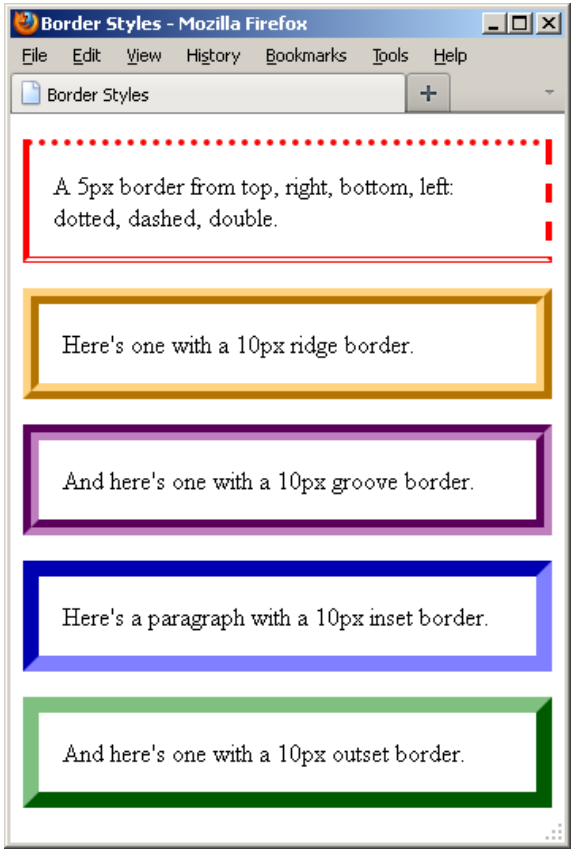
Border Styles

RIGHT BORDER STOPS AND STARTS FOR EACH ENTRY

FOOTER'S TOP BORDER IN DIFFERENT COLOR



Border Styles in Firefox



To Define the Border Style

Type **border-style: *type***

- Where *type* is none, dotted, dashed, solid, double, groove, ridge, inset, or outset.

To Set the Width of the Border

Type **border-width: *n***

- Where *n* is the desired width, including abbreviated units.

To Set the Color of the Border

Type **border-color: *color***

- Where *color* is a color name, hex value, or RGB, HSL, RGBA, or HSLA color.

Set One or More Border Properties with a Shortcut

Type **border**.

- If desired, type **-top, -right, -bottom**, or **-left** to limit the effect to a single side.
- If desired, type ***-property***
  - Where *property* is style, width, or color, to limit the effect to a single property.

Type **:** (a colon).

Type appropriate values.

- If skipped above, specify any or all border properties.
- If you specified a property type, use an accepted value for just that property.

Offsetting Elements in the Natural Flow

Each element has a natural location in a page's flow.

- Relative positioning: Moving the element with respect to this original location.
- The surrounding elements are not affected at all.

To Offset Elements within the Natural Flow

Type **position: relative;** (don't forget the semicolon; the space is optional).

Type **top**, **right**, **bottom**, or **left**.

Then type ***:v***, where *v* is the desired distance that you want to offset the element from its natural location, expressed either as an absolute or relative value or as a percentage.

If desired, repeat step 2 for additional directions, separating each property/value pair with a semicolon as usual.

Positioning Elements Absolutely

Elements in Web page generally flow in the order in which they appear in the HTML source code.

Take elements out of the normal flow—and position them absolutely—by specifying their precise position with respect to the body or nearest positioned ancestor element.

To Position Elements Absolutely

Type **position: absolute;**

If desired, type **top**, **right**, **bottom**, or **left**.

Then type **: *v***,

- Where *v* is expressed as desired distance you want to offset the element from its ancestor or as percentage of ancestor.

If desired, repeat step for additional directions, separating each property/value pair with a semicolon.

If desired, add position: relative to *ancestor* element.

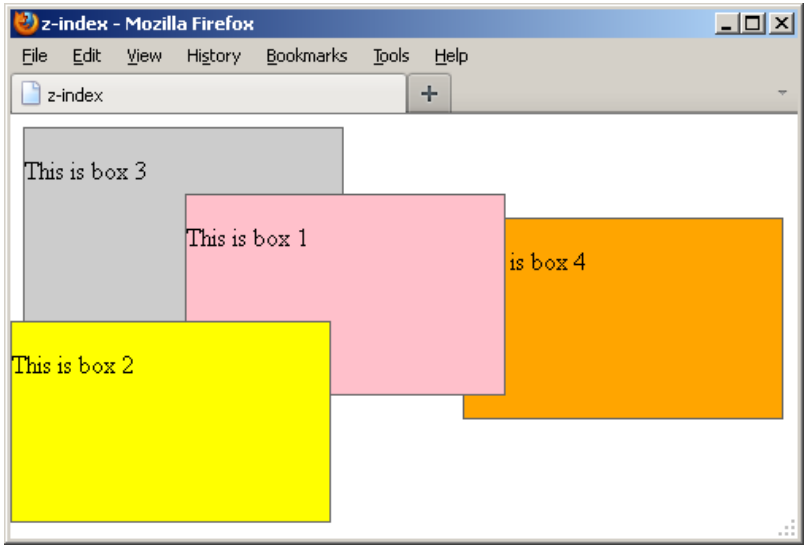
- Skip this step and the element will be offset with respect to the body.

Positioning Elements

When positioned elements have overlapped, you can choose which element should display on top.

Absolutely (or relatively or fixed) positioned element with the highest z-index number always shows on top, regardless of where it appears in the order of the HTML.

Positioned Boxes



Positioned boxes are stacked from highest z-index down to lowest.

- Order of HTML doesn't matter.
- Third box is lowest because it's in normal document flow.

To Position Elements

Type **z-index: *n***,

- Where *n* is a number that indicates the element's level in the stack of positioned objects.

Determining How to Treat Overflow

Overflow property: Controls area outside element's box when element cannot be contained in its box.

To display a single line of images regardless of the browser width, set height of ul element that contains the list of images to the height of the largest images and then set overflow to hidden.

To restrict the viewable area of images to one line but allow visitors to access them all via a scroll bar when they spill to multiple lines, use overflow: auto; in conjunction with the same height as before.

To Determine how the Browser Should Treat Overflow

Type **overflow:**.

Type:

- **visible** to expand the element box so that its contents fit. This is the default option.
- **hidden** to hide any contents that don't fit in the element box.
- **scroll** to always add scroll bars to the element so that the visitor can access the overflow if they so desire.
- **auto**, to have scroll bars appear only when necessary.

Aligning Elements Vertically

Align elements in many different ways to make them look neater on the page.

Images are aligned by default to the bottom of the line.

Aligning Images on page

ALIGNED BY DEFAULT AT BOTTOM

licensed under the Creative Commons Attribution-NonCommercial-NoDerivs L
send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Franci

ALIGNED TO MIDDLE OF LINE

licensed under the Creative Commons Attribution-NonCommercial-NoDerivs L
send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Franci

To Position Elements Vertically

Type **vertical-align:**

Type:

- **baseline** to align element baseline with parent baseline.
- **middle** to align middle of element with middle of parent.
- **sub** to position element as subscript of parent.
- **super** to position element as superscript of parent.
- **text-top** to align top of element with top of parent.

To Position Elements Vertically

Type:

- **text-bottom** to align bottom of element with bottom of parent.
- **top** to align top of element with top of tallest element on line.
- **bottom** to align bottom of element to bottom of lowest element on line.
- A percentage of the line height of the element, which may be positive or negative.

Changing the Cursor

Normally, the browser takes care of the cursor shape for you, using an arrow most of the time, a pointing finger to highlight links, as well as some others. CSS lets you take the reins

To Change the Cursor

Type **cursor:**

Type **pointer** for the cursor that usually appears over links, default for an arrow, crosshair, move, wait, help, text, or progress.

Or type **auto** to get cursor usually appears in that situation.

Or type ***x*-resize** to get double-sided arrow, where *x* is the cardinal direction one of the arrows should point—that is, n (north), nw (northwest), e (east), and so on.

Displaying and Hiding Elements

Display property:

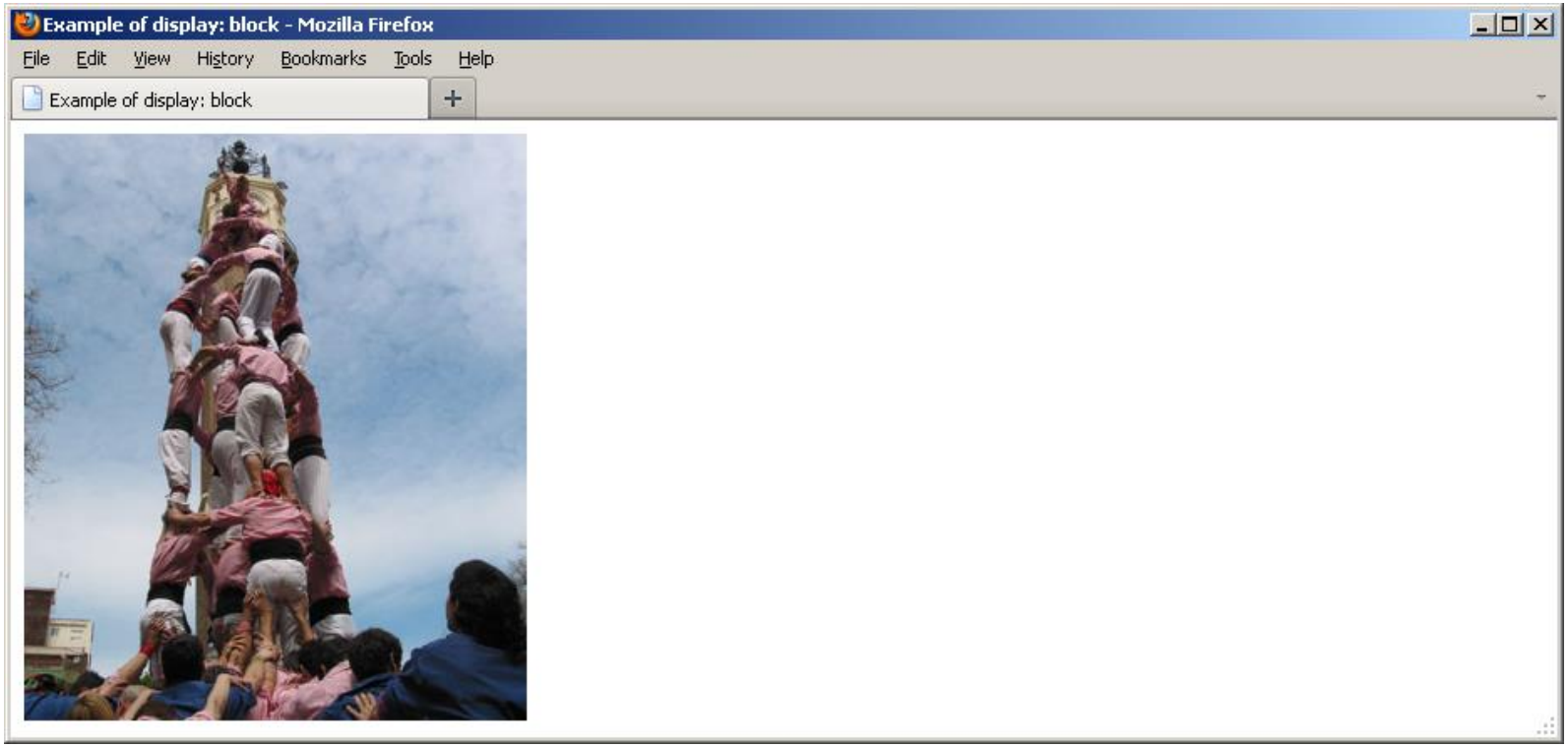
- Can override an element's natural display type.
  - Change it from inline to block or vice versa.
  - Inline-block: hybrid display type that allows an element to appear on same line as other content while otherwise behaving like block-level element.
- Display property is also used to prevent an element and its content from occupying any visual space in the page.

Visibility property controls whether an element is visible.

- Unlike display property, when you hide an element with visibility, a blank space shows where the element would be.

Img Element Default Display Style is Inline

Change Images to Display as Block



Set Display:none to Have No Display

Empty Space Remains with Visibility: hidden

To Specify how Elements Should Display

In your style sheet rule, type **display:**

Type block to display element as block-level.

- Like starting a new paragraph.

Or type **inline** to display the element as inline.

- Not like starting a new paragraph.

Or type **inline-block** to display element as inline but with block-level characteristics.

- Means you can assign element properties on all four sides.

Or type **none** to hide the given element and completely remove it from the document flow.

To Control an Element's Visibility

In your style sheet rule, type **visibility:**

Type **hidden** to make the element invisible without removing it from the document flow.

Or type **visible** to reveal the element.

- When we set hide to have no display…no trace of the second image remains.
- When we remove the display: none; declaration from the hide class and change the visibility property to hidden…an empty space remains where the hidden image used to be.