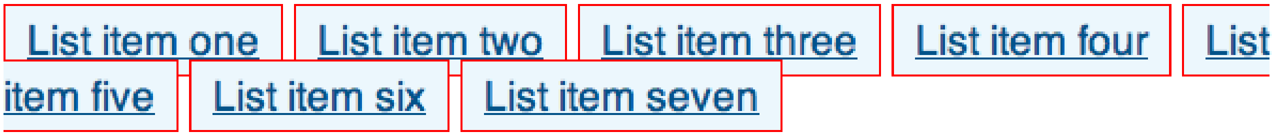
Inline elements and padding

Date: 19 November 2006
Author: Russ Weakley

Have you ever tried to add padding to `<li>` elements that have been set to "display: inline"? Did you find that the padding seemed to be rendering in an unusual way?

In the example below, the `<li>` elements have been set to "display: inline", and then 1em of padding has applied to all sides. Unfortunately, the padding on the top and bottom of the `<li>` elements seems to have been ignored causing the elements to overlap each other.



List item one List item two List item three List item four List item five List item six List item seven

To understand why this is happening, we need to look at the different ways that block level and inline elements treat properties such as width, height, padding and margins.

Block level elements

The W3C's CSS2 spec defines block level elements as "elements of the source document that are formatted visually as blocks".

In other words, block level elements are normally displayed as blocks with line breaks before and afterwards.

Examples of block level elements

Element	Description
`<address>`	information on author
`<blockquote>`	long quotation
`<button>`	push button
`<caption>`	table caption
`<dd>`	definition description
`<del>`	deleted text

`<div>`	generic language/style container
`<dl>`	definition list
`<dt>`	definition term
`<fieldset>`	form control group
`<form>`	interactive form
`<h1>`	heading
`<h2>`	heading
`<h3>`	heading
`<h4>`	heading
`<h5>`	heading
`<h6>`	heading
`<hr>`	horizontal rule
`<iframe>`	inline subwindow
`<ins>`	inserted text
`<legend>`	fieldset legend
`<li>`	list item
`<map>`	client-side image map
`<noframes>`	alternate content container for non frame-based rendering
`<noscript>`	alternate content container for non script-based rendering
`<object>`	generic embedded object
`<ol>`	ordered list
`<p>`	paragraph
`<pre>`	preformatted text
`<table>`	table
`<tbody>`	table body
`<td>`	table data cell

`<tfoot>`	table footer
`<th>`	table header cell
`<thead>`	table header
`<tr>`	table row
`<ul>`	unordered list

Inline elements

The W3C’s CSS2 spec defines inline elements as "elements of the source document that do not form new blocks of content; the content is distributed in lines".

So, inline content is displayed with no line breaks before or afterwards.

Examples of inline elements

Element	Description
`<a>`	anchor
`<abbr>`	abbreviated form
`<acronym>`	acronym
`<b>`	bold text style
`<bdo>`	I18N BiDi over-ride
`<big>`	large text style
`<br>`	forced line break
`<button>`	push button
`<cite>`	citation
`<code>`	computer code fragment
`<del>`	deleted text
`<dfn>`	instance definition
`<em>`	emphasis
`<i>`	italic text style
`<iframe>`	inline subwindow

`<img>`	Embedded image
`<input>`	form control
`<ins>`	inserted text
`<kbd>`	text to be entered by the user
`<label>`	form field label text
`<map>`	client-side image map
`<object>`	generic embedded object
`<q>`	short inline quotation
`<samp>`	sample program output, scripts, etc.
`<select>`	option selector
`<small>`	small text style
`<span>`	generic language/style container
`<strong>`	strong emphasis
`<sub>`	subscript
`<sup>`	superscript
`<textarea>`	multi-line text field
`<tt>`	teletype or monospaced text style
`<var>`	instance of a variable or program argument

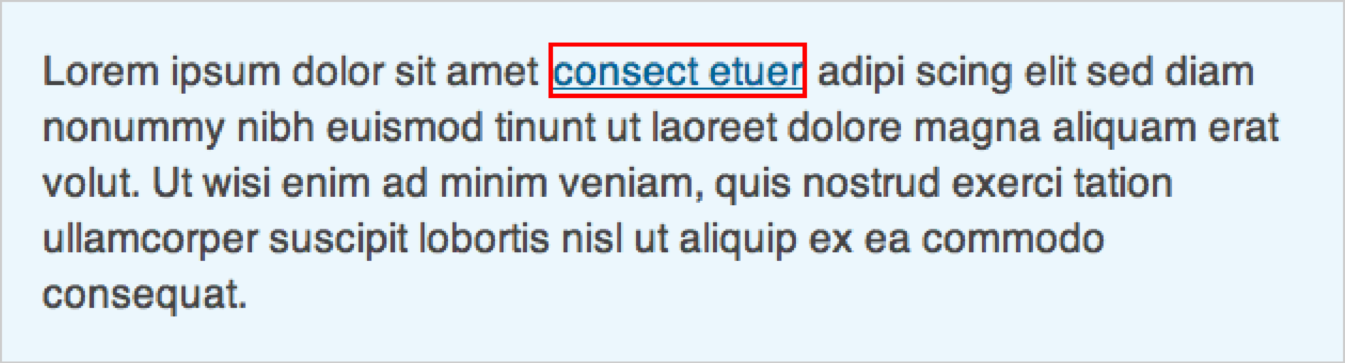
Dimension – a key difference between block and inline elements

If you try to add dimension to an inline element, some properties will be applied, some properties will be partially applied and others will not be applied at all. The most noticable properties are width, height, margin and padding.

Inline elements and width

The W3C’s CSS2 spec states that for Inline, non-replaced elements, "the ‘width’ property does not apply".

In the example below, a width of 200px has been applied to the inline `<a>` element. As you can see, it has no affect on the surrounding content:

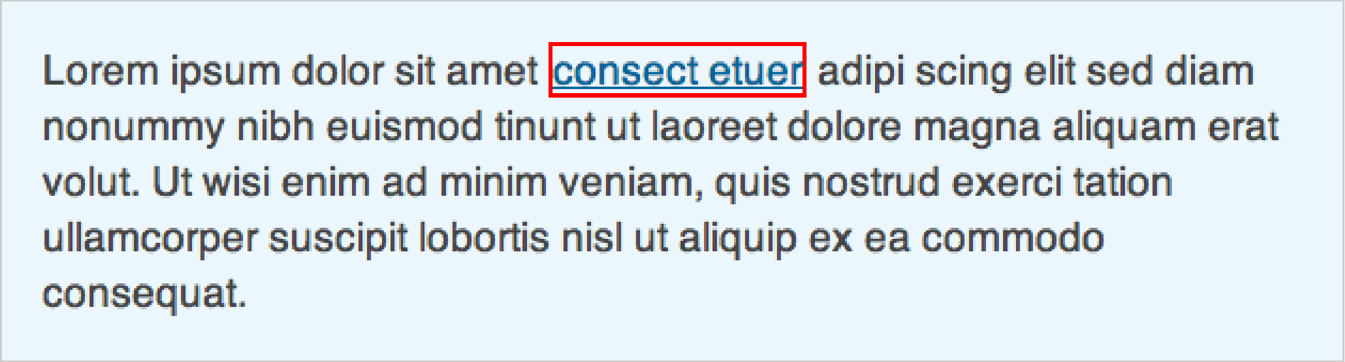


Lorem ipsum dolor sit amet consect etuer adipi scing elit sed diam nonummy nibh euismod tinunt ut laoreet dolore magna aliquam erat volut. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Inline elements and height

The W3C’s CSS2 spec states that for Inline, non-replaced elements, "the ‘height’ property doesn’t apply, but the height of the box is given by the ‘line-height’ property".

In the example below, a height of 50px has been applied to the inline `<a>` element. As you can see, it has no affect on the surrounding content:

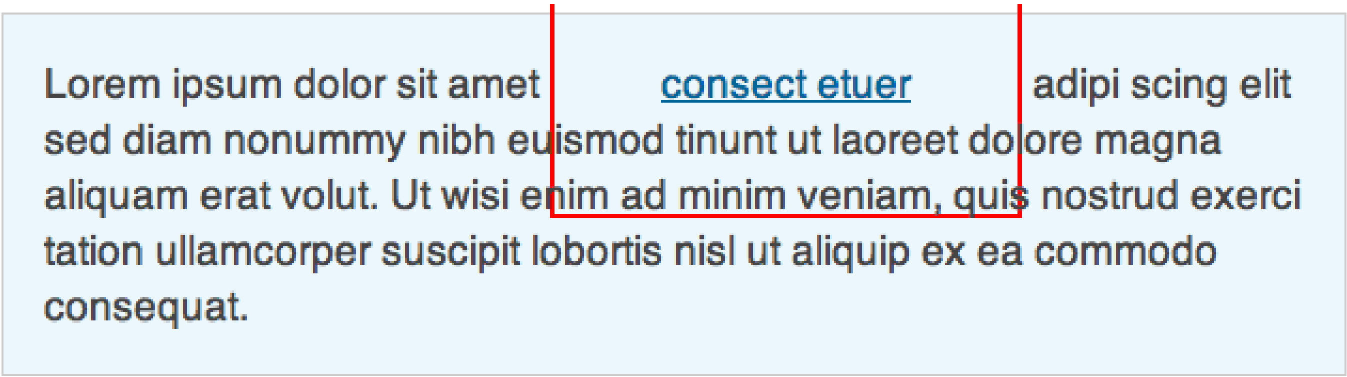


Lorem ipsum dolor sit amet consect etuer adipi scing elit sed diam nonummy nibh euismod tinunt ut laoreet dolore magna aliquam erat volut. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Inline elements and padding

While padding can be applied to all sides of an inline element, only left and right padding will have an effect on surrounding content.

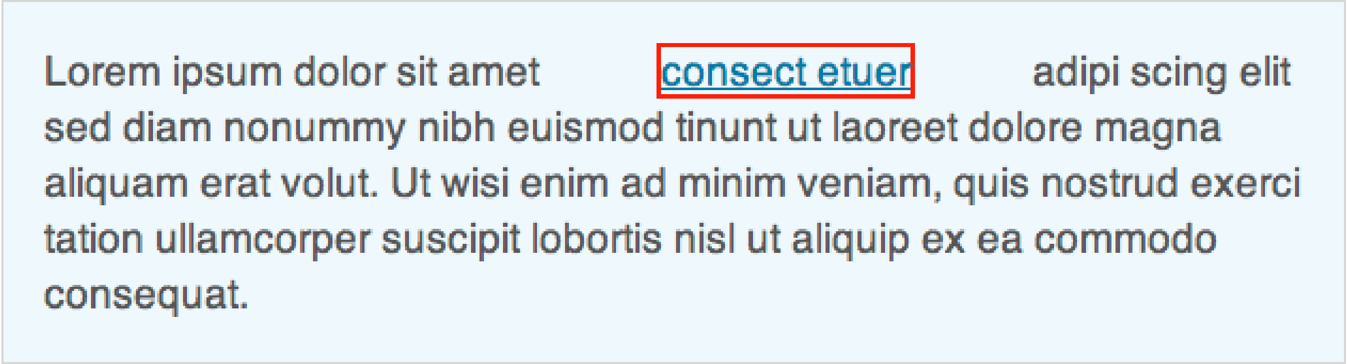
In the example below, 50px of padding has been applied to all sides of the `<a>` element. As you can see, it has an affect on the content on each side, but not on content above or below:



Lorem ipsum dolor sit amet consect etuer adipi scing elit sed diam nonummy nibh euismod tinunt ut laoreet dolore magna aliquam erat volut. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Inline elements and margins

Margins operate in the same way as padding on inline elements. In the example below, 50px of margin has been applied to all sides of the `<a>` element. While the left and right edges are effected, the content above and below are not:

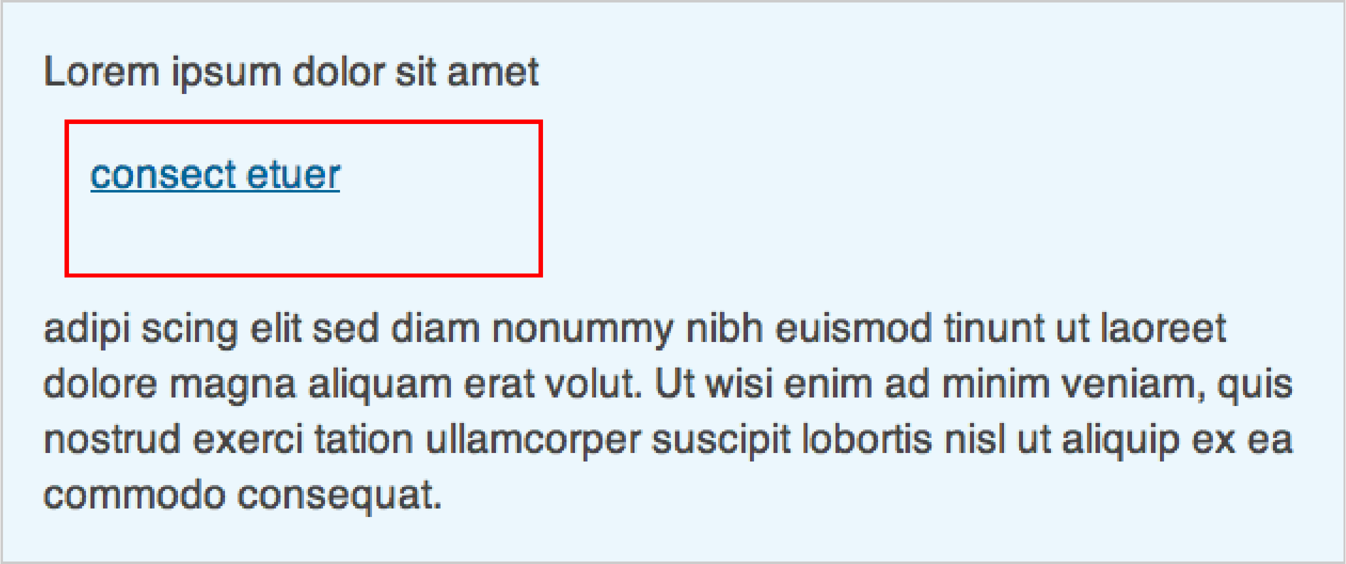


Lorem ipsum dolor sit amet consect etuer adipi scing elit sed diam nonummy nibh euismod tinunt ut laoreet dolore magna aliquam erat volut. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Changing an element’s “display” property from inline to block

It is possible to change the display property of an inline element to “block”. This will give it a block level appearance without changing it’s actual structure.

For example, the `<a>` element below has been set to “display: block”. As soon as this occurs, properties like width, height, margin and padding are applied is if it were a block level element.



Lorem ipsum dolor sit amet

consect etuer

adipi scing elit sed diam nonummy nibh euismod tinunt ut laoreet dolore magna aliquam erat volut. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Changing an element’s “display” property from block level to inline

You can also change block level elements so that they display inline. If an `<li>` element is set to “display: inline”, width, height, padding and margin will immediately operate as they do for any other inline element.

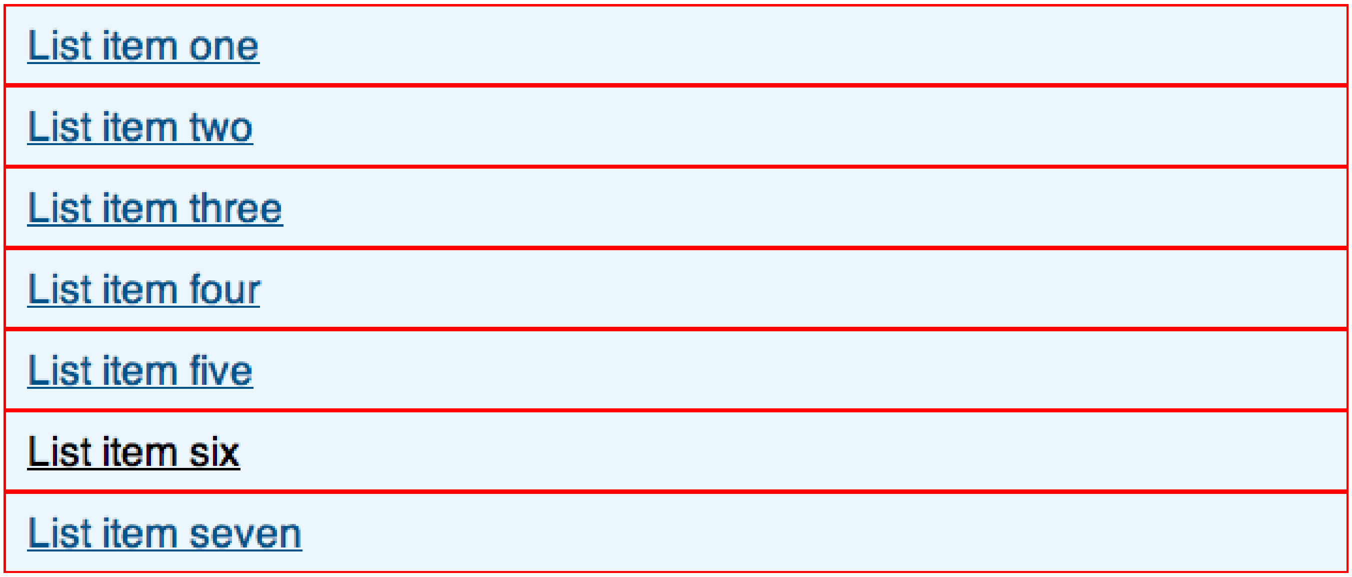
This is what caused the `<li>` element to ignore top and bottom padding in our original example.

Overcoming padding issues for `<li>` elements set to “inline”

There are many ways to overcome the padding issue shown above. Here is one:

Step 1

Remove any reference to “display: inline” and allow the `<li>` elements to return to their natural state – “display: block”. Padding is now applied to all sides of the element, but the elements sit in a vertical stack.



List item one

List item two

List item three

List item four

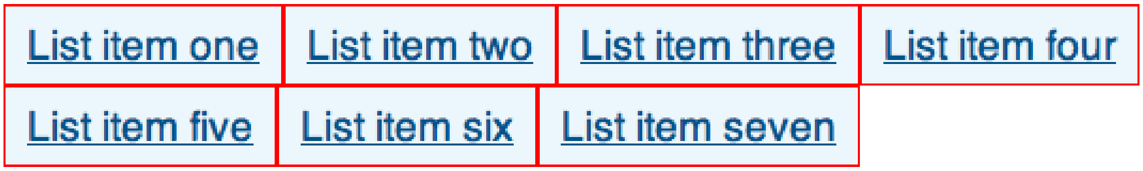
List item five

List item six

List item seven

Step 2

Set the `<li>` element to “float: left” and give it a width – in this case 8em has been used. This will allow the list items to sit beside each other. If there is not enough room for all list items to sit next to each other, those that do not fit will move down and sit below.



List item one List item two List item three List item four

List item five List item six List item seven