

The following is a summary of the implementation status of the various proposals made by [Additional Requirements for Bidi in HTML and CSS](#).

Contents:

- [2.1 Support bidi isolation of inline element content](#)
- [2.2 Support auto-direction](#)
- [2.3 Support reporting the chosen direction of <input> and <textarea> in form submissions](#)
- [2.4 Support option for images to be flipped horizontally in RTL](#)
- [3.1
 should serve as a bidi separator](#)
- [3.2 Line breaks should serve as bidi separators inside <pre>, <textarea>, and script dialog text](#)
- [3.3 Block elements as bidi separators](#)
- [3.4 Script dialog text direction](#)
- [3.5 <title> should support the dir attribute](#)
- [3.6 title and alt attribute text direction](#)
- [3.7 <option> should support the dir attribute and be displayed accordingly both in the dropdown and after being chosen](#)
- [3.8 <input type="text"> and <textarea> should support interoperable "set direction" functionality](#)
- [3.9 When an input value is remembered, its direction should be remembered too](#)
- [3.10 The rendering of numbering or bullets in a list should be independent of the direction of individual elements](#)
- [3.11 A page's overall vertical scrollbar should be on the "end" side relative to the user agent chrome direction](#)
- [3.12 The vertical scrollbar of an element below <body> should be on the "end" side relative to the element's direction](#)

[Notes](#)

2.1 [Support bidi isolation of inline element content](#)

- [CSS Writing Modes Level 3 module added](#) “isolate” and “isolate-override” values to unicode-bidi property.
 - [Bug 613149](#) filed on Mozilla, fixed.
 - [Bug 712600](#) filed on Mozilla, fixed.
 - [Bug 717811](#) filed on Mozilla.
 - [Bug 774335](#) filed on Mozilla, fixed.
 - [Bug 50912](#) filed on WebKit, fixed.
 - [Bug 73164](#) filed on WebKit, fixed.
 - [Bug 74396](#) filed on WebKit, fixed.
 - [Bug 74399](#) filed on WebKit, closed as working as intended after spec changed.

- [Bug 89746](#) filed on WebKit, fixed.
- [Bug 109624](#) filed on WebKit.
- [Bug 124146](#) filed on WebKit, closed as dup of 109624 above.
- [Bug 10807](#) filed on HTML5.
 - Proposes new `ubi=on|off|` attribute, mapping to `unicode-bidi:isolate` for `ubi=on|`.
 - **Instead, [HTML5 added a new bdi element](#)**, which gets `unicode-bidi:isolate` in the default stylesheet, and whose `dir` attribute value is `auto` by default. The output element, previously added to HTML5, is also to get `unicode-bidi:isolate` in the default stylesheet.
 - *Should the output element, which is likely to be used mostly for numbers, really be isolated, with no way to override it in HTML?*
 - [Bug 613149](#) filed on Mozilla, fixed.
 - [Bug 712603](#) filed on Mozilla, fixed.
 - [Bug 50913](#) filed on WebKit, fixed.
 - [Bug 68773](#) filed on WebKit, fixed.
 - [Bug 70373](#) filed on WebKit, fixed.
- [Bug 20085](#) filed on HTML5.
 - Asks to update the description of `bdi` to stay in sync with the CSS definition of `unicode-bidi:isolate`, which changed to accommodate Unicode bidi isolates by leaking forced paragraph breaks out of the isolate.
- [Bug 17670](#) filed on HTML5.
 - Asks to reflect change in CSS spec from “`bidi-override isolate`” to “`isolate-override`”.
 - [Fixed](#).
 - Bug closed.

2.2 [Support auto-direction](#)

- [CSS Writing Modes Level 3 module added](#) “`plaintext`” value to `unicode-bidi` property in order to allow per-paragraph auto-direction, primarily for use on `textarea` and pre elements.
 - [Bug 50949](#) filed on WebKit, fixed.
 - [Bug 65074](#) filed on WebKit, fixed.
 - [Bug 71189](#) filed on WebKit.
 - [Bug 296847](#) filed on Chromium.
 - [Bug 71194](#) filed on WebKit, fixed.
 - [Bug 280709](#) filed on Chromium, fixed.
 - [Bug 73310](#) filed on WebKit, fixed.
 - [Bug 78594](#) filed on WebKit.
 - [Bug 277687](#) filed on Chromium, fixed.
 - [Bug 84543](#) filed on WebKit.
 - [Bug 276642](#) filed on Chromium, fixed.
 - [Bug 662288](#) filed on Mozilla, fixed.

- [Bug 726420](#) filed on Mozilla, fixed.
 - [Bug 746987](#) filed on Mozilla, fixed.
- [Bug 10808](#) filed on HTML5.
 - Proposes new “auto” value for the dir attribute, and a new attribute, autodir=first-strong|any-rtl|plaintext. Elements with dir=auto to be isolated by default (via separately-proposed ubi attribute, see 2.1 above).
 - [HTML5 added](#) a new “auto” value for the dir attribute, **but not autodir**. The effect of dir=auto is to set the unicode-bidi CSS property to “plaintext” for <textarea> and <pre> elements, to “bidi-override isolate” for <bdo> elements, and to “isolate” otherwise. Either way, it estimates a direction according to the UBA method applied to element’s text content (while skipping over descendant elements with an explicit dir attribute, as well as bdi, script and style elements), and sets the CSS direction property accordingly.
 - Bug Closed
 - [Bug 548206](#) filed on Mozilla, fixed.
 - [Bug 50916](#) filed on WebKit, fixed.
 - [Bug 63903](#) filed on WebKit, fixed.
 - [Bug 71188](#) filed on WebKit, fixed.
- [Bug 13482](#) filed on HTML5.
 - Regrets that the “auto” value’s specification bases the directionality on descendant text nodes, and thus does not give sufficient consideration to <input> and <textarea> elements. Proposes that:
 - For input and textarea elements with dir=auto, the directionality should be determined not by the element’s descendants, but by the element’s current value, and should change as the latter changes.
 - For other elements, the text node descendants under a descendant textarea element should be excluded, just like those in script and style elements.
 - [HTML5 changed](#) as requested.
 - Bug Closed
 - [Bug 65428](#) filed on WebKit, fixed.
- [Bug 11734](#) filed on HTML5.
 - Proposes that the special effects of dir=auto on <pre> and <textarea> be documented in the section dir=auto, not simply implied by the default stylesheet.
 - **Proposal accepted**, [adding a note](#) that “For textarea and pre elements, the heuristic is applied on a per-paragraph level.”
 - Bug Closed.

2.3 [Support reporting the chosen direction of <input> and <textarea> in form submissions](#)

- [Bug 10809](#) filed on HTML5.
 - Proposes new submitdir attribute (boolean) for input and textarea elements, which would add a control to the submission whose name is that of the element

suffixed with “_dir”, and whose value is the element’s computed direction at the time of submission.

- [HTML5 added](#) a new attribute with roughly the proposed semantics **but**:
 - i. **Limited it to textareas and inputs of type “text” and “search”.**
 - ii. **Called it dirname. The problem with the name “submitdir” is that most people apparently take it to mean the plausible “directory you submit to” (i.e. similar to “action” - someone even suggested renaming it “actiondir”).**
 - iii. **Required that it be given a value.**
- Reopened:
 - i. There has to be something better than the "dirname" name. How about "addDirection"? - **Rejected: too long, ddd.**
 - ii. Would it be possible to allow no value, e.g. to use the name value suffixed with "_dir"? - **Rejected: feature is already too complicated.**
- Bug Closed
- [Bug 65542](#) filed on WebKit, fixed.
- [Bug 675943](#) filed on Mozilla.

2.4 [Support option for images to be flipped horizontally in RTL](#)

- [CSS3 Images changed](#) to allow an optional “ltr” or “rtl” keyword in the syntax of the image() notation, as suggested above. This is somewhat narrower than the original suggestion to cover all of <image>, which would have included gradients, but is sufficient to address the core need.
- Since no browser has implemented it, image() notation was at one point removed from CSS3 Images with plans to move it to CSS4 Images.
- After an appeal from the W3C i18n WG, image() notation has been [returned to CSS3 Images](#), but is still marked as "at-risk".
 - [Bug 703217](#) filed on Mozilla.
 - [Bug 72584](#) filed on WebKit.
- The ltr and rtl keywords have now been [dropped from CSS3 images](#) due to a design issue. The feature has been deferred to Level 4 (but probably with a different syntax than originally suggested). The [design issue](#) is that it is not clear that there is a use case for specifying different modes (non-flipping, ltr, rtl) for different images in the fallback chain. That the syntax allows this may indicate that it is suboptimal.
- No bug has been filed on HTML5 because “directional” images seem to be part of the presentation, not content, and should be done in CSS, not HTML. Furthermore, there is a way to accomplish the desired conditional flip with CSS rules like .rtl-flip:rtl {transform:scaleX(-1)} that can be applied to HTML elements.

3.1 [
 should should serve as a bidi separator](#)

- [Bug 10828](#) filed on HTML5.
 - Proposes that
 should be a bidi paragraph separator, at least by default.
 - [HTML5 changed](#) to define
 as a bidi paragraph separator.

- The information in [Additional Requirements for Bidi in HTML and CSS](#) turns out to be out-of-date. Although IE7 continued to treat
 as a bidi separator, IE8 and and IE9, *when in standards mode*, treat
 as bidi whitespace, as per the HTML4 spec.
 - *Reopened, advising of the development, and without making a specific recommendation.*
 - [Bug 229367](#) filed on Mozilla years ago, fixed.
 - [Bug 23124](#) filed on WebKit, fixed.
- [CSS 2.1 test added](#) to check support for LINE SEPARATOR and PARAGRAPH SEPARATOR with Unicode semantics in pre element. AFAIK, they are not considered whitespace and thus are not collapsed regardless of the white-space CSS property value, so they should act the same outside a pre element too. Once browsers pass these tests, they will finally start treating   as a line break that does not constitute a bidi paragraph break, i.e. a soft line break.
 - [Bug 53203](#) filed on WebKit.
 - [Bug 215700](#) filed on Mozilla years ago and still active.
- [Bug 11211](#) filed on HTML5.
 - Proposes new bidibreak=hard|soft attribute controlling behavior of
 in descendant elements, defaulting to hard on root element.
 - **Rejected, given the lack of incontrovertibly good use cases for “soft”, and that LINE SEPARATOR will hopefully start getting supported and thus cover the use cases that have been given.**
 - *Close bug?*

3.2 Line breaks should serve as bidi separators inside <pre>, <textarea>, and script dialog text

- [Bug 10812](#) filed on HTML5.
 - Proposes that in <textarea> and <pre>, line breaks should constitute UBA paragraph breaks.
 - **Rejected because the semantics of characters are controlled entirely by CSS, which defers to Unicode, which already defines newlines etc. as paragraph breaks, so Mozilla and Opera are just buggy.**
 - Nevertheless, [HTML5 added](#) a note that newlines constitute UBA paragraph breaks in <pre>.
 - Separate bug filed to get similar note for <textarea>, see below.
 - Bug Closed
 - [Bug 263359](#) filed on Mozilla years ago, fixed.
- [Bug 11502](#) filed on HTML5
 - Proposes to add a note for <textarea> similar to [the one for <pre>](#).
 - [HTML5 changed](#) as requested
 - Bug closed.
- [Bug 10810](#) filed on HTML5.
 - Proposes supporting the LINE SEPARATOR (U+2028) and PARAGRAPH

SEPARATOR (U+2029) characters in <textarea> and <pre> with the semantics assigned to them by the Unicode standard.

- **Rejected because whitespace collapse and the semantics of characters are controlled entirely by CSS, which defers to Unicode.**
- [CSS 2.1 test added](#) to check support for LINE SEPARATOR and PARAGRAPH SEPARATOR with Unicode semantics in pre element.
- Bug Closed
- [Bug 53203](#) filed on WebKit.
- [Bug 215700](#) filed on Mozilla years ago.
- [Bug 10813](#) filed on HTML5.
 - Proposes that in script dialog text, line breaks should constitute UBA paragraph breaks.
 - **Rejected because these are entirely Unicode strings without markup, so their semantics should be entirely defined by Unicode.**
 - Nevertheless, [HTML5 added](#) as an example of required Unicode support in script dialogs that LINE FEED must constitute a UBA paragraph break.
 - Bug Closed
 - [Bug 607640](#) filed on Mozilla. Closed as duplicate of [Bug 263359](#) (now fixed), since Mozilla implements dialogs as an element with preformatted whitespace.
- [Bug 10811](#) filed on HTML5.
 - Proposes supporting the LINE SEPARATOR (U+2028) and PARAGRAPH SEPARATOR (U+2029) characters in script dialog text with the semantics assigned to them by the Unicode standard.
 - **Rejected because these are entirely Unicode strings without markup, so their semantics should be entirely defined by Unicode.**
 - Bug Closed.
 - [Bug 65615](#) filed on WebKit.
 - No need to file a bug on Mozilla, which renders dialogs using an element with preformatted whitespace, so it's covered by [bug 215700](#).

3.3 [Block elements as bidi separators](#)

- [Bug 10814](#) filed on HTML5.
 - Proposes that display:block elements, except those taken out of document flow with CSS such as float or position:absolute, should be specified as introducing a UBA paragraph break between the content preceding and following it.
 - **Rejected as being entirely in the purview of the CSS spec.** HTML5 no longer even has the concept of inline and block elements, and explicitly does not define rendering except by reference to CSS, which applies even if the user agent does not implement CSS (it must translate the CSS spec to its own concepts).
 - The [CSS 2.1 spec on anonymous block boxes](#) does seem to require what we are proposing.
 - Bug Closed.
 - [Bug 613157](#) filed on Mozilla, fixed.

- [Bug 10815](#) filed on HTML5.
 - Proposes removing the [HTML 4 requirement](#) that block elements forced to display:inline should effectively get unicode-bidi:embed. Instead, the default value of the ubi attribute for them (see 2.1 above) should be on.
 - **Instead, [HTML5 changed](#) the default stylesheet to specify unicode-bidi:isolate for all “block” elements (in case they get forced to display:inline), where it had previously specified unicode-bidi:embed.** The HTML 4 requirement mentioned above had already been removed from HTML5.
 - Bug Closed
 - [Bug 65617](#) filed on WebKit.
 - [Bug 296863](#) filed on Chromium.
 - [Bug 676245](#) filed on Mozilla, fixed.
 - [Bug 14850](#) filed on HTML5 because the change made in the default stylesheet would only be effective for “block” elements lacking a dir attribute. The rule that any element with a dir attribute (other than “auto”) gets unicode-bidi:embed is considered to have higher specificity, and thus wins. The change needs to be expanded to “block” elements with a dir attribute.
 - i. Updated [bug 65617](#) on WebKit.
 - ii. Updated [bug 676245](#) on Mozilla.
- [Bug 11829](#) filed on HTML5.
 - Originally proposed to make normative the parts of the default stylesheet setting unicode-bidi and direction, but broadened to require a normative definition of the rules for applying the bidi algorithm (paragraphs, base direction, embedded direction, paragraph breaks inside elements with embedded direction) in the absence of CSS.
 - **[HTML5 changed](#) to say that “user agents that do not honor author-level CSS style sheets are nonetheless expected to act as if they applied the CSS rules given in these sections [...] This is especially important for issues relating to the 'display', 'unicode-bidi', and 'direction' properties.”** Furthermore, “supporting the suggested default rendering” has been made into a [conformance class](#).

3.4 [Script dialog text direction](#)

- [Bug 10827](#) filed on HTML5.
 - Proposes that plain text passed by page scripts to whatever services are provided for dialog display without specifying an explicit direction should be displayed according to the UBA's rules P1, P2, and P3, which estimate the direction of each paragraph according to its first strong character.
 - [HTML5 changed](#) in the direction proposed.
 - Bug Closed.
 - [Bug 70806](#) filed on Chromium, fixed on Windows, Mac, not sure about ChromeOS and Linux.
 - [Bug 146533](#) filed on Chromium, fixed.

- [Bug 652550](#) filed on Mozilla, fixed.
- [Bug 10816](#) filed on HTML5.
 - Proposes requiring user agents to implement the Unicode spec re Default Ignorable Code Points ([Unicode Standard version 5.2. Chapter 5](#), section 5.21), *even if the underlying platform does not handle them properly*. In particular, this must be the case for the directional formatting characters in script dialog text, page titles, and tooltips. The HTML author can then use such characters to indicate the direction of all or part of such text without being careful to do so only when such characters are absolutely necessary, since unnecessary formatting characters would not longer be able to cause harm by being displayed incorrectly.
 - **Rejected because if the underlying platform handles them incorrectly, then that's a bug in the underlying platform. Either the platform should be fixed, or the user agent should work around it. If the Unicode spec requiring something isn't enough, then the HTML spec requiring the same thing isn't going to make any difference.**
 - *Actually, we are not asking for the same thing, but for HTML to require user agents to work around Ignorable Code Points bugs in the platform. Reopen? Or is the user agent already required by the Unicode spec to work around platform bugs, since it has overall responsibility for what it displays? i18n WG: do not reopen.*
 - Bug Closed.

3.5 [<title> should support the dir attribute](#)

- [Bug 10817](#) filed on HTML5.
 - Proposes requiring that title element text will be displayed in the element's computed direction.
 - [HTML5 changed](#) as proposed.
 - In addition, [HTML5 changed](#) to require text from elements generally to be rendered in native user interfaces in a manner that honors the directionality of the element from which the text was obtained, mentioning title bars, pop-up menus, and tooltips as particular examples.
 - Bug Closed.
 - [Bug 50961](#) filed on Webkit, fixed.
 - [Bug 27094](#) filed on Chromium
 - [Bug 145418](#) filed on Chromium
 - [Bug 703242](#) filed on Mozilla.

3.6 [title and alt attribute text direction](#)

- [Bug 10818](#) filed on HTML5.
 - Proposes adding two new attributes, `titledir=ltr|rtl|auto` and `altdir=ltr|rtl|auto` that specify the title and alt attributes' direction, respectively. In their absence, the title

- and alt attribute text is to be displayed in the element's computed direction.
 - [HTML5 changed](#) to require text from elements generally, including their attribute values, to be rendered in native user interfaces in a manner that honors the directionality of the element from which the text was obtained, mentioning dialogs, title bars, pop-up menus, and tooltips as particular examples.
 - **Addition of titledir and altdir attributes rejected because they are inelegant (can not add ...lang and ...dir attributes for every attribute with a displayed text value), and cases where the element direction and the title or alt text direction need to be different are rare, and a workaround is available.**
 - Bug Closed.
 - Note: [Chromium](#) already does that.
- [Bug 17814](#) filed on HTML5.
 - Complains that it makes no sense to limit the placeholder attribute to values of the same direction as the <input>. Halfheartedly proposes to always display the placeholder as if it had dir=auto, but would prefer to define a placeholderdir attribute for <input>.
 - The solution offered by the fix for [Bug 17829](#), i.e. to use dir=auto on an input with placeholder, should be good enough for most use cases.
 - Bug Closed.
- [Bug 17829](#) filed on HTML5.
 - Complains that dir=auto has unintended results on user-visible attribute values, but there is no dir=auto for user-visible attribute values.
 - Proposes to solve this as well as [bug 10818](#) and [bug 17814](#) by a new attribute, attribsdir=ltr|rtl|auto. When not specified:
 - i. If the element has dir=auto (explicitly or by default, as is the case for the bdi element), or if the element inherits its directionality from such an element, then the directionality of each of the element's attributes must be computed as if attribsdir="auto" had been specified.
 - ii. Otherwise, the directionality of the element's attributes is the same as the element's directionality.
 - **Adding an attribsdir attribute rejected.**
 - [HTML5 changed](#) so that the user-visible attributes of an element with dir=auto get their own directionality, in which they are to be rendered, based on just their content.
 - **Doing so for elements that lack a dir attribute but inherit their directionality from an element with dir=auto rejected.**
 - [Bug 1169986](#) filed on Mozilla.
 - [Bug 494553](#) filed on Chrome.
 - [Bug 145498](#) filed on WebKit.

3.7 <option> should support the dir attribute and be displayed accordingly both in the dropdown and after being chosen

- [Bug 10819](#) filed on HTML5.

- Proposes specifying that the option element's computed direction will take its dir attribute (explicit or inherited) into account, and will be used to display the option's text in both the dropdown and after being chosen.
- [HTML5 changed](#) to require text from elements generally to be rendered in native user interfaces in a manner that honors the directionality of the element from which the text was obtained. Gives a detailed example for the option element, including the correct rendering in the select.
- Bug Closed.
- [Bug 50969](#) filed on Webkit, verified fixed on Windows, Linux and Mac.
- [Bug 10820](#) filed on HTML5.
 - Proposes specifying that the option element's alignment via CSS or the align attribute will affect its display accordingly in both the drop-down and after being chosen. **The proposal is misguided** because:
 - i. HTML5 no longer includes the align attribute.
 - ii. This is thus purely a CSS matter.
 - iii. Currently, no browser lets the text-align CSS property affect an option's alignment either in the drop-down or after being chosen.
 - iv. If text-align, which is "start" by default, were allowed to influence option's alignment, the results would be undesirable: by default, LTR and RTL entries would have opposite alignment, which in most cases is undesirable.
 - After a series of misunderstandings starting with the above, [HTML5 changed](#) to specify that **"User agents are expected to render the labels in a select in such a manner that any alignment remains consistent whether the label is being displayed as part of the page or in a menu control."**
 - *Close the bug, or reopen to get rid of this change, which (if I understand it correctly) is only violated by Mozilla (which aligns options according to their direction in the drop-down, but not after selection)?*
 - [Bug 614851](#) filed on Mozilla.

3.8 [<input type="text"> and <textarea> should support interoperable "set direction" functionality](#)

- [Bug 10821](#) filed on HTML5.
 - Proposes specifying that when the user agent user interface allows the user to set the direction of <input> and <textarea> elements, it will set the element's dir attribute value accordingly and trigger the oninput event after the dir attribute has been set.
 - [HTML5 changed](#) as proposed. In fact, the change includes that the user agent "should allow the user to change the writing direction of the element". **However, the change is only applied to inputs of type "text" and "search".**
 - Bug Closed
 - [Bug 50952](#) filed on WebKit, fixed.
 - [Bug 675579](#) filed on Mozilla, fixed.

- [Bug 91178](#) filed on Chromium, fixed.
- [Bug 327294](#) filed on Chromium.
- [Bug 113211](#) filed on Chromium, no longer reproduces.
- [Bug 10822](#) filed on HTML5.
 - Proposes recommending that on an OS that has a widespread convention for setting direction, the user agent should support that convention on input and textarea elements.
 - **Proposal rejected because “user agents are naturally incentivised to follow platform conventions, but should not be required to do so, since they may have perfectly good reasons for doing entirely different things. It's out of scope of the specification to require behaviour that is not a factor for interoperability.”**
 - Bug Closed.

3.9 [When an input value is remembered, its direction should be remembered too](#)

- [Bug 10823](#) filed on HTML5.
 - Proposes specifying that whenever a user agent stores a user-provided <input> or <textarea> text value for later use (such as auto-completion), it should also store the nominal direction value the element had when displaying this value. If the user agent later recalls and displays this value, e.g. in an auto-completion dropdown, it should be displayed in its stored direction. If the value is assigned to an element, the element's dir value should be set to its stored direction.
 - Proposal rejected because:
 - i. “We definitely don't want session storage state memory mutating the DOM dynamically.” But form memory already does so when it sets the input/textarea's value/content.
 - ii. “In any case, that's unlikely to ever come up, since the page isn't likely to have a different dir="" attribute than it did when the user filled the form the first time.” But it does, when the user sets the control's direction via the UI provided by the user agent, which sets the control's dir attribute, when filling the form the first time.
 - iii. “The spec doesn't define how any of the form memory stuff should work, so it's not clear to me that it'd be appropriate to go into this much detail.” This is a fair point: it's difficult to be normative in something that the spec really doesn't define very much.
 - Bug reopened with explanations for i and ii above.
 - **Proposal accepted.** [Note added](#) that among the updates that the user agent can make in “aspects of the document and its rendering” on history traversal is that it “can even include updating the dir attribute of textarea elements or input elements whose type attribute is in either the Text state or the Search state, if the persisted state includes the directionality of user input in such controls.” [Note added](#) that “User agents that persist the value of form controls are encouraged to also persist their directionality (the value of the element's dir attribute). This

prevents values from being displayed incorrectly after a history traversal when the user had originally entered the values with an explicit, non-default directionality.”

- Bug closed.
- [Bug 65429](#) filed on WebKit.
- [Bug 675491](#) filed on Mozilla.

3.10 The rendering of numbering or bullets in a list should be independent of the direction of individual elements

- [CSS Text Level 3 module added](#) “match-parent” value to the text-align property. It assigns to the element the same effective alignment as its parent. For example, if we have:

```
<div id="a" dir="rtl" style="text-align:start">  
  <div id="b" dir="ltr" style="text-align:match-parent">...</div>  
</div>
```

then b is aligned left like its parent. Without match-parent it would be aligned right.
 - [Bug 50951](#) filed for Webkit, fixed.
 - [Bug 79914](#) filed for Webkit, fixed.
 - [Bug 645642](#) filed for Mozilla, fixed.
- [Thread](#) started on W3C CSS WG mailing list, with major improvements made during discussion at the CSS WG face-to-face at TPAC 2010.
 - Proposes a new value for list-style-position: outside-parent. The new value would work like the existing list-style-position:outside, except that it would apply the list element's (as opposed to the list item element's) direction to the marker box. The marker box direction would affect both the bidirectional ordering of the marker text and the location of the marker box, which is on the "start" side of its direction. While the default value of list-style-position would continue to be “outside” for CSS2 compatibility, it is proposed that the default stylesheet would specify {list-style-position:outside-parent; text-align: match-parent} for li elements.
 - Positively received by the editor-to-be of [CSS3 Lists](#). The list-style-position:outside-parent value was positively received by the CSS WG face-to-face at TPAC 2010. Putting text-align:match-parent for li elements into the default stylesheet was only proposed after TPAC.
- [Bug 10824](#) filed on HTML5.
 - Proposes that while CSS will provide means to control whether list item markers use the list element's computed direction instead of the list item's for the marker's display and positioning, the HTML default should be to use the list element's computed direction. If the CSS default in this respect must differ, the default stylesheet should achieve the desired HTML default behavior.
 - HTML editor has accepted changing the default stylesheet for this purpose, but the CSS WG has yet to approve the precise change to be made. **Describing the default bidi behavior of lists in the HTML spec has been rejected as being entirely in the purview of the CSS spec.** (HTML5 explicitly doesn't define

rendering except by reference to CSS, which applies even if the user agent does not implement CSS (it must translate the CSS spec to its own concepts).

- *Reopen bug once the exact change to the default stylesheet has been approved by the CSS WG.*

3.11 A page's overall vertical scrollbar should be on the "end" side relative to the user agent chrome direction

- [Bug 10825](#) filed on HTML5.
 - Proposes specifying that the user agent window's overall vertical scrollbar should be located independent of the direction of any page element, despite being otherwise controlled by the style of the body element.
 - **Rejected as out-of-scope: "The HTML spec doesn't even require that there be a window in the first place, let alone a scroll bar. Plus, user interface decisions are explicitly left up to user agents since they represent quality-of-implementation issues and not interoperability issues."**
 - Bug Closed.
- It appears likely that a proposal to add this to the CSS spec would also be rejected, since "CSS explicitly mentions only a *scrolling mechanism* and considers out-of-scope how that mechanism is implemented. It could be scroll bars, or a little panner map in the corner, arrow buttons along each edge of the box, or a joystick control with no on-screen representation, or something else."
- *Since scrollbar position is legitimately out of scope for both CSS and HTML, create a new W3C document of recommendations for user agent implementors to cover such issues?*

3.12 The vertical scrollbar of an element below <body> should be on the "end" side relative to the element's direction

- [Bug 556363](#) filed on Mozilla, and has been fixed. However, does not have the special case for frames/iframes.
- [Bug 619963](#) filed on Mozilla for scrollbars in iframes.
- [Bug 54623](#) filed on WebKit, fixed.
- [Bug 84026](#) filed on WebKit.
- [Bug 250514](#) filed on Chromium.
- [Bug 85856](#) filed on WebKit, closed as Chromium-specific.
- [Bug 128500](#) filed on Chromium.
- [Bug 10826](#) filed on HTML5.
 - Proposes specifying that the vertical scrollbar of an element below the body element, and of the body element of a document being displayed in a frame or iframe, should be on the "end" side relative to the element's direction.
 - **Rejected as out-of-scope: "Where the scroll bars render is a presentational concern, and out of scope for HTML. HTML barely mentions scroll bars, in fact (and only in relation to historical APIs)."**

- Bug Closed.
- It appears likely that a proposal to add this to the CSS spec would also be rejected, as explained for 3.11 above.
- *Since scrollbar position is legitimately out of scope for both CSS and HTML, create a new W3C document of recommendations for user agent implementors to cover such issues?*

Notes

1. Mozilla [bug 613154](#) lists all the Mozilla bugs opened so far that deal with [Additional Requirements for Bidi in HTML and CSS](#).
2. WebKit [bug 50910](#) is the tracking bug for the WebKit project.