

Web for All

- **Developer support**
 - Standards developers & user agent implementers should enable local needs
- **Language enablement**
 - In-country users and other experts tell us their requirements, to pass on to developers.
- **Author support**
 - Content authors and users learn how to use locally-adapted features.





Developer support

Build standards & applications that support a global Web

Guangzhou, China

RTL on the Web

- Arabic script (incl. Persian, Urdu, Uighur, Kashmiri, etc.)
- Hebrew
- Thaana
- Rohingya
- Syriac, Mandaic, Assyrian, etc.
- N'Ko, Adlam, Mende Kikakui
- Etc...

Bidi in HTML5

W3C Working Group Note

Additional Requirements for Bidi in HTML & CSS

W3C Working Group Note 21 July 2015

This version: <http://www.w3.org/TR/2015/NOTE-html-bidi-20150721/>
Latest published version: <http://www.w3.org/TR/html-bidi/>
Previous version: <http://www.w3.org/TR/2010/WD-html-bidi-20100304/>
Editor: Aharon Lanin, Google
Richard Ishida, W3C
Copyright © 2013-2015 W3C® MIT, ERCIM, Keio, Beihang! W3C liability trademark and document use rules apply.

Abstract

Authoring a web app that needs to support both right-to-left and left-to-right interfaces, or to take as input and display both left-to-right and right-to-left data, usually presents a number of challenges that make it an especially laborious and bug-prone task. Some of these are due to browser bugs, but some can be traced back to the specification of the bidirectional aspects of a given HTML, or CSS feature. And some of these challenges can be greatly simplified by adding a few strategically placed new HTML, and CSS features.

This document was used to work through and communicate recommendations made to the HTML and CSS Working Groups for some of the most repetitive pain points. It is being published now for the historical record, and to capture some of the thinking that lay behind the evolution of the specifications and to help people in the future working on bid issues understand the history of the decisions taken. Notes have been added to give a brief summary of what was actually implemented in the HTML or CSS specifications.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](http://www.w3.org/TR/) at <http://www.w3.org/TR/>.

This document contains initial proposals for features to be added to HTML to support bidirectional text in languages such as Arabic, Hebrew, Persian, Thai, Urdu, etc., and describes the eventual solutions that were adopted by HTML5. This is a W3C Draft produced by the Internationalization Working Group, part of the [W3C Internationalization Activity](#). The Working Group expects to advance this Working Draft to Working Group Note. Please send comments on this document to public-idin-bidi@w3.org (publicly archived).

This document was published by the [Internationalization Working Group](#) as a Working Group Note. If you wish to make comments regarding this document, please send them to public-idin-bidi@w3.org (subscribe/archives). All comments are welcome.

Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

TOP RATED RESTAURANTS

Aroma - 3 reviews פיצה סגולה - 5 reviews

TOP RATED RESTAURANTS

Aroma - 3 reviews פיצה סגולה - 5 reviews

Bidi isolation

Bidi in HTML5

W3C Working Group Note

Additional Requirements for Bidi in HTML & CSS

W3C Working Group Note 21 July 2015

This version: <http://www.w3.org/TR/2015/NOTE-html-bidi-20150721/>
Latest published version: <http://www.w3.org/TR/html-bidi/>
Previous version: <http://www.w3.org/TR/2010/WD-html-bidi-20100304/>
Editor: Aharon Lanin, Google
Richard Ishida, W3C
Copyright © 2013-2015 W3C® MIT, ERCIM, Keio, Beihang! W3C liability trademark and document use rules apply.

Abstract

Authoring a web app that needs to support both right-to-left and left-to-right interfaces, or to take as input and display both left-to-right and right-to-left data, usually presents a number of challenges that make it an especially laborious and bug-prone task. Some of these are due to browser bugs, but some can be traced back to the specification of the bidirectional aspects of a given HTML, or CSS feature. And some of these challenges can be greatly simplified by adding a few strategically placed new HTML, and CSS features.

This document was used to work through and communicate recommendations made to the HTML and CSS Working Groups for some of the most repetitive pain points. It is being published now for the historical record, and to capture some of the thinking that lay behind the evolution of the specifications and to help people in the future working on bid issues understand the history of the decisions taken. Notes have been added to give a brief summary of what was actually implemented in the HTML or CSS specifications.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](http://www.w3.org/TR/) at <http://www.w3.org/TR/>.

This document contains initial proposals for features to be added to HTML to support bidirectional text in languages such as Arabic, Hebrew, Persian, Thai, Urdu, etc., and describes the eventual solutions that were adopted by HTML5. This is a W3C Draft produced by the Internationalization Working Group, part of the [W3C Internationalization Activity](#). The Working Group expects to advance this Working Draft to Working Group Note. Please send comments on this document to public-idin-bidi@w3.org (publicly archived).

This document was published by the [Internationalization Working Group](#) as a Working Group Note. If you wish to make comments regarding this document, please send them to public-idin-bidi@w3.org (subscribe/archives). All comments are welcome.

Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

The names of these states in Arabic **الكويت** and **مصر, البحرين** respectively.

The names of these states in Arabic **الكويت** and **مصر, البحرين** respectively.

string metadata



يتم تحقيق ذلك بإضافة العنصر المضمن `bdi`.

```
<textarea dir="auto"></textarea>
```



في HTML5 يتم تحقيق ذلك بإضافة العنصر المضمن `.bdi`.



string metadata



W3C @w3c · 6h

Call for Review: CSS Fonts Module Level 3 is a W3C Proposed Recommendation ift.tt/2vJzUld



1



5



✉

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

⋮

string metadata

W3C @w3c · 6h
Call for Review: CSS Fonts Module Level 3 is a W3C Proposed Recommendation [ift.tt/2vJzUId](#)

r12a @r12a · 6h
في HTML5 يتم تحقيق ذلك بإضافة العنصر المضمن .bdi

W3C Developers @w3cd devs · 6h
The first #CSS #Houdini specification to reach #CandidateRecommendation: CSS Painting Level 1 [w3.org/TR/2018/CR-css...](#) - this is a major step in the evolution of the extensibility of Web browsers

string metadata

W3C @w3c · 6h
Call for Review: CSS Fonts Module Level 3 is a W3C Proposed Recommendation [ift.tt/2vJzUId](#)

r12a @r12a · 6h
تم تحقيق ذلك بإضافة العنصر المضمن .bdi في HTML5

W3C Developers @w3cd devs · 6h
The first #CSS #Houdini specification to reach #CandidateRecommendation: CSS Painting Level 1 [w3.org/TR/2018/CR-css...](#) - this is a major step in the evolution of the extensibility of Web browsers



Bidi support for strings

Strings on the Web: Language and Direction Metadata
W3C Editor's Draft 22 July 2020

This version: <https://w3c.github.io/string-meta/>
Latest published version: <https://www.w3.org/TR/string-meta/>
Latest editor's draft: <https://w3c.github.io/string-meta/>

Editor:
Addison Phillips (Amazon.com)
Richard Ishida (W3C)

Participate:
[GitHub w3c/string-meta](#)
[File a bug](#)
[Commit history](#)
[Pull requests](#)

Copyright © 2007-2020 W3C® (MIT, ERCIM, Keio, Behring). W3C liability, trademark and permissive document license rules apply.

Abstract
This document describes the best practices for identifying language and base direction for strings used on the Web.

Status of This Document
This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at <https://www.w3.org/TR/>.

We welcome comments on this document, but to make it easier to track them, please raise separate issues for each comment, and point to the section you are commenting on using a URL.

This document was published by the Internationalization Working Group as an Editor's Draft.

GitHub Issues are preferred for discussion of this specification.

Publication as an Editor's Draft does not imply endorsement by the W3C Membership.

This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the W3C Patent Policy. The group does not expect this document to become a W3C Recommendation. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

X تصميم و إنشاء موقع الويب: CSS و HTML

✓ تصميم و إنشاء موقع الويب: CSS و HTML

spec reviews

github.com/w3c/i18n-request/projects/1

Early review opportunities	Review requested	In review	Awaiting comment resolution	Completed
DCAT Rev > FPWD (2018-05-08) #57 opened by aphillips FPWD	CSS Writing Modes Level 4 > Ongoing #520 opened by aphillips css-writing-modes	Verifiable Credentials Data Model 1.0 > 2018-09-20 #578 opened by aphillips FPWD needs-review	HTML 5.3 > 2018-05-25 #539 opened by aphillips LC html	ORDL Information Model (+ more) > 2017-05-15 #397 opened by aphillips LC
CSS Logical Properties and Values Level 1 (16 May 2017) #427 opened by r12a FPWD	CSS Painting API Level 1 > 2018-10-09 #581 opened by aphillips LC css-paint-api	CSS Generated Content 2016-06-20 #283 opened by r12a FPWD	CSS Align > 2017-06-30 #422 opened by aphillips LC	Basic Card Payment, 2016-04-21 #269 opened by r12a WD
Resource Timing 2016-04-26 #284 opened by r12a FPWD			TTML2 > 2017-09-30 #428 opened by aphillips CR	Gamepad API > 2018-06-27 #569 opened by LWatson CR
CSS Object Model 2016-03-23 #285 opened by r12a FPWD			Accessibility (A11Y) Requirements for People with Low Vision > 2106-03-17 #285 opened by r12a FPWD	WebVTT #399 opened by r12a CR LC
CSS Ruby #264 opened by r12a WD			Web Authentication, > 2016-08-16 #268 opened by r12a WD	Device and Sensors Reviews (6 docs) > 2017-12-31 #512 opened by aphillips CR
Media Stream Track Content Hints > FPWD (2018-07-03) #576 opened by aphillips media-capture-and...			Browser Payment API > 2017-02-22 #347 opened by aphillips LC	DOM 4.1 > 2018-03-01 #527 opened by xlq CR
			Review request: changes to CR of DNT (tracking-dnt) #487 opened by r12a CR	Semantic Sensor Network Ontology > Urgent #421 opened by aphillips LC
			UTR #53, Unicode Arabic Mark Ordering Algorithm #494 opened by r12a FPWD	ARIA > 2017-04-30 #322 opened by aphillips LC
			Intersection Observer review #405 opened by LWatson FPWD	UI Events KeyboardEvent key Values, > 2016-11-27 #267 opened by r12a LC
				Screen Orientation

Current spec work for Indic

W3C Internationalization (i18n)
Making the World Wide Web worldwide!

Learn Find Contact Participate Follow

Language enablement issue tracker

This page tracks issues related to language support on the Web. Issues listed may be requests to a local community for information about the behaviour of their writing system, or requests for improvements to either specs or browser implementations.

These issues are also linked to from the [Language enablement index](#).

The sections and items within those sections are ordered by the last modified date of the items in the trocadero repository (not the discussions). The mostly recently changed items appear highest. The link on the left of each row links directly to the issue where the discussion is taking place. The date indicates when the trocadero item was modified, and the right-hand column links to that item in the [i18n-activity github repo](#).

You can filter the list in the URL. Add one or two of the filter names in the right-hand column after `?filter=`. If adding two, separate them with `,`.

There are 15 issues.

Baselines & inline alignment	Apr 4, 2019 666
Issues with ZWJ/ZWNJ which baseline values are required?	

Character encodings	Sep 7, 2018 694
Issues with ZWJ/ZWNJ	

Text alignment & justification	Dec 5, 2019 849
Is full inter-character spacing common for narrow columns in Tamil?	

Text decoration	Jul 20, 2017 465
How are underlines positioned in Indic text?	

Useful links

- Tracker issues on GitHub
- International text layout and typography index
- Review tracker

Filter results

Click to filter items by type:

- type-info-request
- spec-type-issue
- browser-type-bug
- Clear filters

Click to filter items by layout group:

- arireq (Africa)
- arieq (Arabic/Persian)
- cree (Chinese)
- eqliq (Ethiopic)
- eurreq (Europe)
- hneq (Hebrew)
- ireq (Indic)
- jreq (Japanese)
- kreq (Korean)
- lreq (Latin)
- mreq (Mongolian)
- searreq (Southeast Asia)
- treq (Tibetan)
- Clear filters

Current spec work for Indic

Styling initials	Dec 6, 2019 808
Should drop initial styling highlight aytam alone?	
Versals in Tamil, Telugu, Malayalam, etc	Aug 23, 2019 737
alignment of initial-letter for South Asian scripts without hanging baseline	Aug 23, 2019 735

undefined	Mar 14, 2019 849
Positioning of prescript vowel signs in non-conjunct clusters	
Does the ievN ABNF work for consonant clusters that don't form conjuncts?	Sept 7, 2018 581
When does the ABNF work for Tamil consonant clusters?	Jul 14, 2017 406

Line breaking	Mar 26, 2019 663
Mid-word-break	

undefined	Apr 21, 2020 886
Hijr calendar abbreviation for Urdu	
Handling western & Indian digits	Sep 7, 2018 593

Lists, counters, etc	Dec 5, 2019 826
Are alphabetic counter styles common in Tamil content?	
Support css-counter-styles-3	Jul 5, 2018 170

[Make list](#) [List info requests](#) [List spec issues](#) [List browser bugs](#) [List close](#)



Much work done to support RTL & bidi

Guidelines for developers

In-depth research and advice

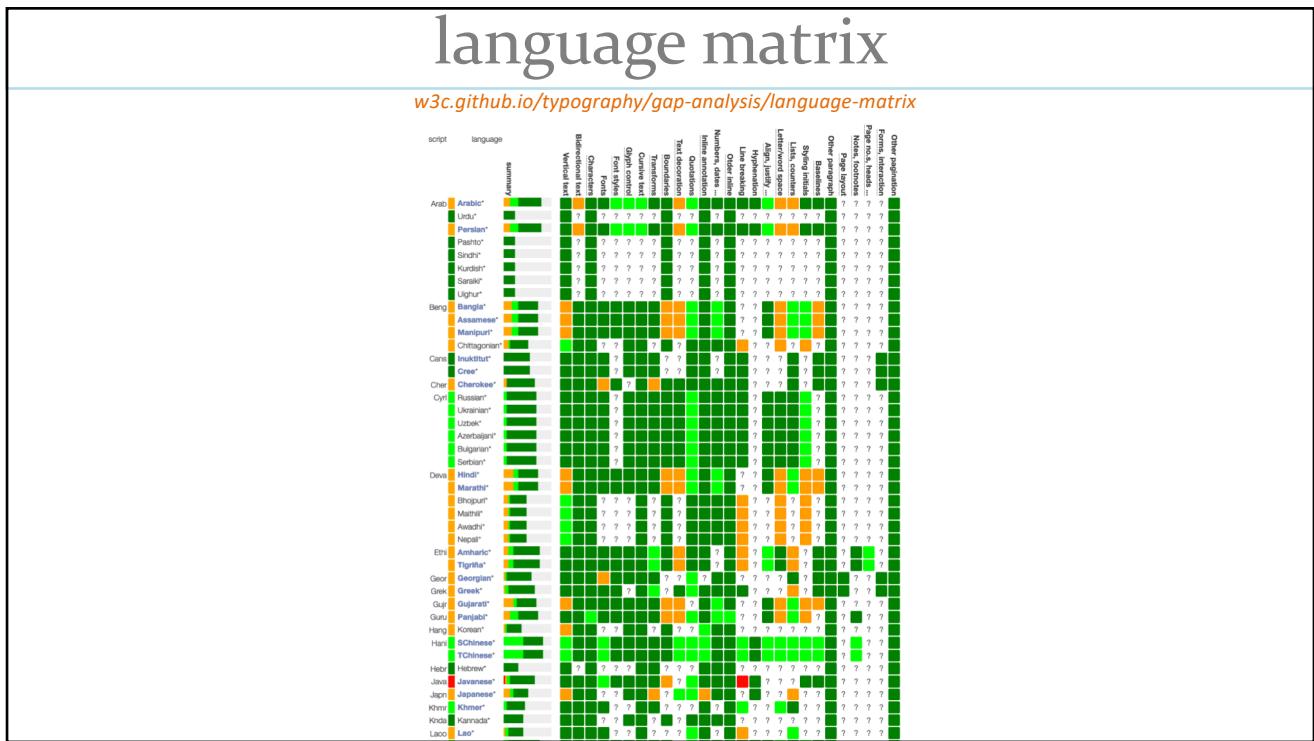
Work still very much in progress



Language Enablement

Understand where the gaps are for users of the Web.

Muscat, Oman



language matrix

TABLE OF CONTENTS

1. Introduction
- 1.1 Work flow
- 1.2 Prioritization

2. Text direction
- 2.1 Vertical text
- 2.2 Bidirectional text

3. Characters and phrases
- 3.1 Characters & encoding
- 3.2 Fonts
- 3.3 Font styles, weight, etc
- 3.4 Glyph shaping and positioning
- 3.5 Cursive text
- 3.6 Baseline, line-height, etc
- 3.7 Transforming characters
- 3.8 Grammar, word segmentation & selection
- 3.9 Inline features & punctuation
- 3.10 Text decoration
- 3.11 Quotations
- 3.12 Inline notes & annotations
- 3.13 Data formats & numbers

4. Lines and Paragraphs
- 4.1 Line-breaking
- 4.2 Hyphenation
- 4.3 Text alignment & justification
- 4.4 Letter spacing
- 4.5 Lists, counters, etc.
- 4.6 Styling initials

5. Page & book layout
- 5.1 General page layout & progression
- 5.2 Footnotes, endnotes, etc.
- 5.3 Page headers, footers, etc.
- 5.4 Forms & user interaction

6. Other
- 6.1 Culture-specific features
- 6.2 What else?

Bengali Gap Analysis

W3C First Public Working Draft 16 June 2020

This version: <https://www.w3.org/TR/2020/WD-beng-gap-20200616/>
Latest published version: <https://www.w3.org/TR/beng-gap/>
Latest editor's draft: <https://w3c.github.io/p/gap-analysis/beng-gap/>

Editor: Richard Ishida (W3C)

Participate:

- [GitHub w3c/wip](#)
- [File a bug](#)
- [Commit history](#)
- [Pull requests](#)

Copyright © 2018-2020 W3C® MIT EBCM Keio Béihang. W3C liability, trademark and permissive document license rules apply.

Abstract

This document describes and prioritises gaps for the support of languages using the Bengali script on the Web and in eBooks. In particular, it is concerned with text layout. It checks that needed features are supported in W3C specifications, in particular HTML, and CSS and those relating to digital publications. It also checks whether the features have been implemented in browsers and eReaders. This is a preliminary analysis.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at <https://www.w3.org/TR/>.

This document describes and prioritises gaps for the support of languages using the Bengali script on the Web and in eBooks. In particular, it is concerned with text layout. It checks that needed features are supported in W3C specifications, in particular HTML, and CSS and those relating to digital publications. It also checks whether the features have been implemented in browsers and eReaders. This document complements the document [Bengali Layout Requirements](#), which describes the requirements for areas where gaps appear. It is linked to from the language matrix that tracks Web support for many languages.

The editor's draft of this document is being developed by the India International Program Task Force, part of the W3C Internationalization Interest Group. It is published by the Internationalization Working Group. The end target for this document is a Working Group Note.

This document was published by the Internationalization Working Group as a First Public Working Draft.

[GitHub Issues](#) are preferred for discussion of this specification.

§ 4.5 Lists, counters, etc.

The CSS Counter Styles specification describes a limited set of simple and complex styles for counters to be used in list numbering, chapter heading numbering, etc. The rules plus more counter styles totalling around 120 for over 30 scripts are listed in the document [Ready-made Counter Styles](#). Do these cover your needs? Are the details correct? Are there other aspects related to counters and lists that need to be addressed? [See available information](#) or [check for currently needed data](#).

#68 Customised counter styles are unavailable

Bengali CSS counter style is defined in the document [Ready-made Counter Styles](#). Same is also defined in the [CSS Counter Styles specification](#), the other relies on the user-defined mechanism specified in that spec in order to be applied.

Needs work for advanced level support.

§ 4.6 Styling initials

Does the browser or eReader correctly handle special styling of the initial letter of a line or paragraph, such as for drop caps or similar? How about the size relationship between the large letter and the lines alongside? Where does the large letter anchor relative to the lines alongside? Is it normal to include initial quote marks in the large letter? Is the large letter really a syllable? etc. Are all of these things working as expected? [See available information](#) or [check for currently needed data](#).

#69 Incorrect segmentation for styling initials

Because of the problems associated with grapheme cluster boundaries (see above), first-letter selection in CSS doesn't work well for conjuncts. For example, Chrome fails to style the whole conjunct in শাস্তির when using :first-letter in a selector, and styles only the শ instead of শা. Similar is the case of Internet Explorer, it only styles শ. This is problematic for many words in a script such as Bengali, and forces the content author to use explicit spans rather than the proper mechanism for selecting initial letter.

CSS uses the concept of 'typographic character unit', rather than grapheme cluster, in its specs with the explanation that these cases are beyond the scope of the grapheme cluster concept and that implementations should provide appropriate support. In addition, a modification to the concept of grapheme cluster is [currently in development](#) at the Unicode Consortium, which is likely to resolve the problem for a script like Bengali.

In addition, the alignment of styled initial-letter character glyphs with the rest of the text is not clearly specified or implemented.

See requirements at: [Indic Layout Requirements](#)

language enablement requirements

**Bengali Layout Requirements
(Preliminary Editor's Draft)**

W3C Editor's Draft 21 August 2020

This version: <http://w3c.github.io/iip/bengali/>
Latest published version: <https://www.w3.org/TR/bengreq/>
Latest editor's draft: <http://w3c.github.io/iip/bengali/>
Editor: Richard Ishida (W3C)
Github: [repository](#)

Copyright © 2019-2020 W3C® (MIT, ERCIM, Keio). W3C liability, trademark and permissive document license rules apply.

Abstract

This document describes requirements for the layout and presentation of text in languages that use the Bengali script when they are used by Web standards and technologies, such as HTML, CSS, Mobile Web, Digital Publications, and Unicode.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the [W3C technical reports index](#) at <https://www.w3.org/TR/>.

This early draft has not yet been through any review process. Please do not rely on the contents.

This document describes the basic requirements for Bengali script layout and text support on the Web and in e-books. These requirements provide information for Web technologies such as CSS, HTML, and digital publications about how to support users of Bengali scripts. Currently the document focuses on Bengali as used for the Bangla language. The information here is developed in conjunction with a document that summarizes gaps in support on the Web for Bengali.

The editor's draft of this document is being developed by the [India International Program Task Force](#), part of the [W3C Internationalization Interest Group](#). It will be published by the [Internationalization Working Group](#). The end target for this document is a Working Group Note.

language enablement requirements

4.1 Grapheme boundaries Needs review

The basic unit for working with Bengali text is the orthographic syllable, i.e. one consonant or a sequence of consonants with hasant between, plus optional additional combining characters (such as vowel-signs).

In Bengali an orthographic syllable that forms a conjunct should be treated as an indivisible unit of text for most editing operations. Figure 1 shows a Bengali word with a conjunct at the end, and the expected segmentation.

Figure 1: Expected minimal units (right) during segmentation of the word ঝি+ঞ্জিল্লি

যি
ঞ্জিল্লি → যি+ঞ্জিল্লি

5.2 Counters Needs review

Counters are used to number lists, chapter headings, etc.

Bengali uses a numeric counter style, based on the decimal model, and using the standard Bengali digits '০' '১' '২' '৩' '৪' '৫' '৬' '৭' '৮' '৯' in a decimal pattern.

1 ⇒ ১ 2 ⇒ ২ 3 ⇒ ৩ 4 ⇒ ৪ 8
11 ⇒ ১১ 22 ⇒ ২২ 33 ⇒ ৩৩ 44 ⇒ ৪৪
111 ⇒ ১১১ 2222 ⇒ ২২২

Figure 2: Expected segmentation of the word ঝি+ঞ্জিল্লি when there is no conjunct.

যি
ঞ্জিল্লি → যি+ঞ্জিল্লি

Figure 3: Note that in Bengali an orthographic syllable may be longer than a Unicode grapheme cluster, if it forms a conjunct. Figure 3 shows a Bengali word with a conjunct at the end, and the segmentation that would result from applying Unicode grapheme clusters only.

যি
ঞ্জিল্লি → যি+ঞ্জিল্লি

Figure 4: Examples of counter values using the Bengali numeric counter style.

text layout index

w3c.github.io/typography/

TABLE OF CONTENTS

- Introduction
- Text direction
 - Vertical text
 - Bidirectional text direction
- Characters & Phrases
 - Fonts
 - Font styles, weight, etc.
 - Glyph shaping & positioning
 - Cursive text
 - Transforming characters
 - Text segmentation & selection
 - Punctuation
 - Text decoration
 - Quotations
 - Inline notes & annotations
 - Numbers & digits
 - Other inline features
- Lines & paragraphs
 - Line breaking
 - Hyphenation
 - Text alignment & justification
 - Word & letter spacing
 - Lists, counters, etc.
 - Styling initials
 - Baselines & inline alignment
- Layout & styling
 - General page layout and progression
 - Grids & tables
 - Notes, footnotes, etc
 - Page headers, footers, etc
 - Forms & user interaction
- Changes Since the Last Published Version

Language enablement index
W3C Editor's Draft 22 November 2019

This version: <https://w3c.github.io/typography/>
 Latest published version: <https://www.w3.org/TR/typography/>
 Latest editor's draft: <https://w3c.github.io/typography/>

Editor: Richard Ishida (W3C)
Participate:
[GitHub w3c/typography](#)
[File a bug](#)
[Commit history](#)
[Pull requests](#)

Copyright © 2019 W3C®. MIT. ERMIC, Inc. Beihang. W3C liability, trademark and permissive document license rules apply.

Abstract
 This document points browser implementers and specification developers to information about how to support typographic features of scripts or writing systems from around the world, and also points to relevant information in specifications, to tests, and to useful articles and papers. It is not exhaustive, and will be added to from time to time.

Status of This Document
 This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at <https://www.w3.org/TR/>.

The information in this document helps to link users and developers so that browsers can better support typographic needs around the world. It is expected that this document will be constantly updated, as new material becomes available or comes to our attention.

NOTE
 To make it easier to track comments, please raise separate issues or emails for each comment, and point to the section you are commenting on using a URL, for the dated version of the document.

This document was produced by the Internationalization Working Group as an Editor's Draft.
 GitHub Issues are preferred for discussion of this specification.
 Publication as an Editor's Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.
 This document was produced by a group operating under the W3C Patent Policy. The group does not

4.2 Hyphenation [5](#)

Some scripts don't use hyphenation, those that do have particular rules about how it should be applied that are typically language-specific.

Requirements

- Indic Layout Requirements: [Hyphenation](#)
- Latin Layout Requirements: [Hyphenation-The Classical Rules of Hyphenation and Pagination](#)

GitHub resources

- All issues
- Requests for information
- Spec issues
- Useful discussions
- Type samples

Spec links

- CSS3 Text: [Hyphenation: the hyphens property](#)

Tests

- CSS Text: [Hyphens](#)

Gap analysis

- Dutch
- Hungarian
- Javanese

questions about Indic scripts

Internationalization (i18n)
Making the World Wide Web worldwide!

Learn Find Contact Participate Follow

Language enablement issue tracker

This page tracks issues related to language support on the Web. Issues listed may be requests to a local community for information about the behaviour of their writing system, or requests for improvements to either specs or browser implementations.

These issues are also linked to the [Language enablement index](#).

There are 11 issues.

Text decoration
 How are underlines positioned in Arabic text? [Jul 20, 2017](#) | 467

Vertical text
 Vertical text handling in Unicode [Sep 7, 2018](#) | 596

Numbers, etc
 Hij calendar abbreviation for Urdu. [Apr 21, 2020](#) | 886
 Why are bidi categories of Arabic-Indic & Eastern Arabic numbers different? [Mar 8, 2019](#) | 647

Hyphenation
 Hyphenation in Arabic script writing systems [Mar 8, 2019](#) | 645

Cursive text
 Are vertically-misaligned characters joined? [Nov 19, 2019](#) | 82
 Isctext! Should zero width space break Arabic shaping? [Apr 25, 2019](#) | 695
 Correct Spacing for LAM before Non-Arabic and BEH Before Number [Sep 7, 2018](#) | 595

Useful links
 Tracker issues on Github
 International text layout and typography index
 Review tracker

Filter results
 Click to filter items by type:
 type-info-request
 spec-type-issue
 browser-type-bug
 Clear filters
 Click to filter items by layout group:
 arireq (Africa)
 alreq (Arabic/Persian)
 cire (Chinese)
 elreq (Ethiopic)
 eurreq (Europe)
 hirreq (Hebrew)
 ireq (Indic)
 jreq (Japanese)
 kreq (Korean)
 lreq (Latin)
 mreq (Mongolian)
 sealreq (Southeast Asia)
 treq (Tibetan)
 Clear filters

networks

w3c.github.io/iip/

This screenshot shows the main repository page for `w3c/iip`. It includes the README file which describes the project's purpose of documenting gaps and requirements for support of Indic languages on the Web and in eBooks. The Issues section lists several open issues related to various scripts like Bengali, Devanagari, and Tamil.

This screenshot shows the Issues page with a filter applied to show only open issues. The list includes items such as 'Devanagari: 3.11 Numbers, Dates - providing numbers and dates in Devanagari or Latin scripts should be handled by CLDR locales' and 'Devanagari: 3.9.3 Embedded quotation marks in multi-script text - clarify use case, gather specimens'.

Language Enablement

A photograph of a person in traditional Middle Eastern attire (kandura and agal) sitting on a polished floor in a grand, arched hallway with intricate architectural details and hanging lanterns, likely a mosque.

Language matrix

Gap-analysis

Layout requirements

Expert networks

Need more experts!



Author support

Help people create content in their own language, or create content that will be localised.

Wakan, Oman

articles

 Internationalization (i18n) Activity
Making the World Wide Web wonderful!

[Home](#) [Resources](#) [Techniques](#) [Topics](#) [News](#) [Groups](#) [About](#)

i18n site search:

RSS Feeds

Articles, best practices & tutorials

You can also find resources using the [Technique index](#) and [Topic index](#), which provide more fine-grained access to information.

> [አማርኛ](#) [Bengali](#) [Dansk](#)
[Deutsch](#) [Eλληνικά](#) [Español](#) [Français](#) [ଓଡ଼ିଆ](#)
[Magyar](#) [日本語](#) [မြန်မာဘ်](#) [Nederlands](#) [Polski](#)
[Português](#) [Português-BR](#) [Română](#) [Pycckий](#)
[Svenska](#) [ไทย](#) [Українська](#) [简体中文](#)

Getting Started

[Getting Started with the W3C i18n site](#)
[Introducing Character Sets and Encodings](#)
[Language on the Web](#)
[Internationalization Quick Tips for the Web](#)

On this page

[Getting Started](#)
[Characters](#)
[Language](#)
[Markup & text](#)
[Text direction](#)
[Styling & layout](#)
[Forms](#)
[Navigation](#)
[Web addresses](#)
[Cultural issues](#)
[Other](#)

Characters

[Handling character encodings in HTML and CSS \(tutorial\)](#)
[Character encodings for beginners](#)
[Character encodings: Essential concepts](#)
[Choosing & applying a character encoding](#)
[Declaring character encodings in HTML](#)
[Declaring character encodings in CSS](#)
[The byte-order mark \(BOM\) in HTML](#)
[Normalization in HTML and CSS](#)
[Characters or markup?](#)
[Changing an HTML page to Unicode](#)
[Using character escapes in markup and CSS](#)
[Document character set](#)
[Setting the HTTP charset parameter](#)
[Setting charset information in .htaccess](#)
[Checking HTTP Headers](#)
[Checking the character encoding using the validator](#)
[HTML, XHTML, XML, and Control Codes](#)
[Missing characters and glyphs](#)
[Who uses Unicode?](#)
[Migrating to Unicode](#)

Language

[Internationalization \(i18n\) Activity](#)

Getting started

Characters

Language

Markup & text

Text direction

Styling & layout

Forms

Navigation

Web addresses

Cultural issues

vertical text guidelines

W3C Internationalization

Home Resources Techniques Topics News Groups About

Styling vertical Chinese, Japanese, Korean and Mongolian text

Intended audience: CSS3 developers, and anyone who needs guidance on how to produce vertically-oriented text for Chinese, Japanese and Korean using CSS.

Updated 2017-08-13 10:53

This article explains how to use CSS to produce vertical text for languages such as Chinese, Japanese, Korean, and Mongolian. The CSS specification contains a lot of implementation-specific details about styling vertical text, and this article is a summary that current authors need to create the more common features of vertical text. It also compares the theory and practice about what is possible.

Each section explains how you would mark it up your content according to the CSS spec (which is usually simple and straightforward), but then looks at what you currently have to do to achieve the same result in browsers that don't implement the standard property and value names. For more information, see [Browser support](#).

Basic setup

Where it is supported, most of what you need should be achievable by applying the `writing-mode` property to the content that you want to be set vertically.

In Japanese, Chinese and Korean, lines start at the right side of the figure box and progress to the left. Latin script text typically runs down the page, with the letters rotated clockwise, while the Han characters remain upright. Any graphic also remains upright.



Fig. 1. Chinese and Japanese vertical text lines run right to left.

About this article

This article has been reviewed by the W3C Internationalization Working Group and has gone through the process of review. If you have comments, send them using the link near the bottom of the page.

CSS

```
.figure { writing-mode: vertical-rl; }
.upright { text-orientation: upright; }
```

HTML

```
<figure>
  <span class="upright">i</span> は、浅葱の双子の
  兄であり、共犯者だ。
</figure>
```

Fig. 7. Using a fullwidth transform to make Latin letters stand upright.

The letters 'W3C' in the above example initially ran down the page, but applying the fullwidth transform using the following CSS makes them stand upright.



Fig. 7. Using a fullwidth transform to make Latin letters stand upright.

This is appropriate for initials, but is not necessarily useful for all types of upright text, and note especially that this technique only works for Latin characters without accents!

Using fullwidth characters. Another way to achieve this is to just use fullwidth characters, such as W 3 C. These will automatically be displayed upright by default. You don't need any markup in this case.

Of course, this also only works for Latin script text that doesn't include accents (since those are the only letters for which full-width variants exist).

Internationalization W3C

i18n test suite

w3.org/international/tests

W3C Internationalization

Home Resources Techniques Topics News Groups About

Summarized test results: CSS3 Writing Modes, vertical text

Intended audience: users, HTML coders, script developers, CSS coders, Web project managers, and anyone who wants to know whether browsers support the CSS Ruby spec.

Updated 2016-12-02 12:07

These tests check whether user agents correctly apply the `writing-mode` property per the CSS3 spec for the `vertical-rl` and `vertical-rl` values. They are just essential tests. More detailed tests for edge cases and finer aspects of rendering can be found in the CSS test suite.

To see the test, click on the link in the left-most column. To see detailed results for a single test, click on a row and look just above the table. The detailed results show the date(s) the test result was recorded, and the version of the browser tested.

Any dependencies are shown in notes above the table, and notes below the table will usually provide any additional useful information, including an explanation of why a result was marked as "partially successful".

Key:

- green
- yellow
- partially successful

The proprietary test results are for either preferred implementations, using `-webkit-` or `-ms-`, or for the nightly version of Firefox, or for non-standard writing-mode values in Internet Explorer.

If `writing-mode-rl-001` or `writing-mode-rl-001` fails, or either of the corresponding `-prop` tests, the remaining tests for the section can be ignored.

vertical-rl

Basics

Test link	Assertion	Firefox	Chrome	Opera	Safari	Edge	IE	Android	UC
<code>writing-mode: vertical-rl;</code>	writing-mode:vertical-rl will display a line of text vertically.	pass	pass	pass	fail	pass	fail	fail	fail
<code>writing-mode: vertical-rl; line</code>	writing-mode:vertical-rl; line	pass	pass	pass	fail	pass	fail	fail	fail
<code>writing-mode: vertical-rl; wrap</code>	writing-mode:vertical-rl; wrap	pass	pass	pass	fail	pass	fail	fail	fail
<code>writing-mode: vertical-rl; align: center</code>	writing-mode:vertical-rl; align: center	pass	pass	pass	fail	pass	fail	fail	fail

Proprietary syntax

Test link	Assertion	Firefox	Chrome	Opera	Safari	Edge	IE	Android	UC
<code>writing-mode: vertical-rl;</code>	writing-mode:vertical-rl will display a line of text vertically.	pass	pass	pass	pass	pass	pass	pass	pass
<code>writing-mode: vertical-rl; line</code>	writing-mode:vertical-rl; line	pass	pass	pass	pass	pass	pass	pass	pass
<code>writing-mode: vertical-rl; wrap</code>	writing-mode:vertical-rl; wrap	pass	pass	pass	pass	pass	pass	pass	pass
<code>writing-mode: vertical-rl; align: center</code>	writing-mode:vertical-rl; align: center	pass	pass	pass	pass	pass	pass	pass	pass

Glyphs

Test link	Assertion	Firefox	Chrome	Opera	Safari	Edge	IE	Android	UC
<code>writing-mode: vertical-rl; direction: rtl;</code>	By default, writing-mode:vertical-rl will display Arabic characters right-to-left.	pass	pass	pass	fail	pass	fail	fail	fail
<code>writing-mode: vertical-rl; direction: ltr;</code>	By default, writing-mode:vertical-rl will display Arabic characters inside 90° by default.	pass	pass	pass	fail	pass	fail	fail	fail
<code>writing-mode: vertical-rl; direction: rtl; orientation: vertical-rl;</code>	By default, writing-mode:vertical-rl will display Arabic characters progressing up the page.	pass	pass	pass	fail	pass	fail	fail	fail
<code>writing-mode: vertical-rl; direction: rtl; orientation: vertical-rl;</code>	By default, writing-mode:vertical-rl will display Arabic characters using counter-clockwise rotation.	pass	pass	pass	fail	pass	fail	fail	fail

interactive tests

w3.org/international/tests

The screenshot shows the GitHub repository for character phrase tests. An issue titled "ZWSP will not interrupt cursive joining behaviour in Arabic text. #26" is open. The issue details that ZWSP (Zero Width Space) does not interrupt cursive joining in Arabic text. It lists various fonts tested: Gezira, Amiri, Noto Naskh Arabic, Noto Naskh Arabic, Myriad Arabic, Damascus, Blini, Chrome, Schenherazade, Amiri, Noto Naskh Arabic, Al Bayan, and Noto Sans Arabic. It also mentions a website test using Safari. The issue has labels: #26, Arabic, #zwsp, #cursive, #partial, and #weakPartial. The issue was created by r12a on April 6, 2014, and has 0 comments. The repository has 27 issues and 0 pull requests.

techniques index

w3.org/International/techniques/authoring-html

The screenshot shows the Internationalization (i18n) techniques index page. The main heading is "Internationalization techniques: Authoring HTML & CSS". Below it is a sub-heading "Authoring HTML & CSS". A note states: "This page lists links to resources on the W3C Internationalization Activity site and elsewhere that help you author HTML and CSS for internationalization. It is one of several techniques pages." A "Collapse all" link is available. The page lists several sections: "Characters", "Language", "Choosing language tags", and "How to's". Under "Language", there are links for "Getting started", "Declaring the overall language of a page", "Identifying in-document language changes", and "Choosing language tags". The "Choosing language tags" section includes a note: "Use subtags as defined by BCP 47 for language attribute values." and a note: "Use the shortest possible language tag values." Under "How to's", there are links for "Choosing a Language Tag", "Language tags in HTML and XML", "Tagging text with no language", and "Type hints or language codes". The footer includes links for "Learn", "Find", "Contact", and "Follow".

i18n checker

validator.w3.org/i18n-checker

Detailed report

Conflicting character encoding declarations

Explanation

The following character encoding declarations are inconsistent:

- a. <meta charset="iso-8859-1"/>
- b. <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

Browsers will apply precedence rules to determine the character encoding to use for the page, but this may not be the encoding you intended.

What to do

Change the character encoding declarations so that they match. Ensure that your document is actually saved in the encoding you choose.

Further reading

Character encodings explained
Choosing a character encoding
Changing the encoding of a document

[TOP](#)

Issues

- ✖ Multiple encoding declarations using the `meta` tag
- ✖ Content-Language `meta` element used
- ✖ A language attribute value was incorrectly formed
- ✖ A language subtag is invalid
- ✖ A `lang` attribute value did not match an `xml:lang` value when they appeared together on the same tag.
- ⚠ Non-UTF-8 character encoding declared
- ⚠ Non-preferred name used for legacy character encoding
- ⚠ Found Unicode code points for directional controls
- ⚠ Unpaired directional controls found
- ⚠ `b` tags found with no class attribute

 The W3C validators are hosted on server technology donated by HP, and supported by community donations.
[Donate](#) and help us build better tools for a better web.

[Home](#) [About](#) [GitHub](#) [Feedback](#) [Version 2.0.3](#)

Internationalization (I18n) Activity
Making the World Wide Web world wide!

COPYRIGHT © 2013-2016 W3C® (MIT, ERCIM, Keio, BEIRWANG). ALL RIGHTS RESERVED.
W3C LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY.
YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

Author support



[Internationalisation articles](#)

[Techniques index](#)

[Internationalisation test suite](#)

[Internationalization checker](#)

important message

The W3C isn't a Genie in a lamp that solves all your problems. It brings people together & facilitates work, but this is **your Web**.

To produce change we need people **like you** to step up and provide guidance and work through issues.



Punakha, Bhutan

Get involved and help us ensure that it meets local needs around the world.

How to help?

If you know of experts who can help us improve the Web for RTL support and for Indic typography, please let us know.

If you are able, please consider supporting the work through the Internationalization Sponsorship program.

Use and tell people about the work we're doing, and the educational materials we produce to help RTL content development.

W3C Internationalization Initiative

W3C's goal is a Web for All, regardless of language, script or culture. The Web community has made tremendous progress in internationalizing the Web over the last decade. As Web technologies have become more complex, interest increases, as usage scenarios grow, and as new applications such as digital publishing emerge, there remains more to do.

For the Web to truly work for all individuals all over the world, there must be a concerted effort by experts—site designers, developers, and vendors who are active in moving the Web forward. To ensure a rapid response to the growth of the Web, the W3C wants to marshal the resources of the internationalization community around the world and enlist their help in strengthening internationalization support for the Web.

To accelerate progress in this area, the W3C is also looking to supplement the core funding it receives from W3C Member fees so that it can increase in-house resources dedicated to this work.

The internationalization initiative will provide participants and funding to address three main aspects of the internationalization continuum:

- **Language enablement** appeals most directly to stakeholders (e.g., governments, publishers, community groups, etc.) who utilize the language(s) of their country.
- **Developer support** appeals most directly to tech companies that are building the infrastructure for a global Web and supporting W3C standards.
- **Author support** appeals to people creating Web content in their own language, as well as to companies who build or localize Web sites in many languages.

The Web needs your help!
Success in meeting this goal requires participation and funding from language, developer, and author communities, in order to expand the effort over and above what can be achieved with our core funding.



Language enablement
The W3C wants to ensure that local requirements for language support on the Web are met. This is particularly true for the Indian subcontinent where the Devanagari script is in use. The W3C is in an area of particular interest, because the W3C has developed a set of rules for line-breaking & justification, local according to existing principles or according to traditional writer styles, supporting bidirectional text in multiple scripts, and so on. These typographic features are often very different from the Western norm in languages that use writing systems such as Arabic, Chinese, and others.

In Web design and digital publishing, this can be addressed by providing W3C standards for rendering text such as the W3C's CSS3 specification for markup (such as HTML). The goal here is to ensure that the local typographic features that users around the world are used to, and are able to use, are supported correctly in the Web, with long-standing traditions. This requires a clear and detailed overview of where work is needed.

To achieve this goal, the W3C needs to assemble a network of experts in the various Indian languages on the Web, identify gaps, prioritize them, develop requirements, and propose standards to fill those gaps. To do so, it needs to establish a network of experts, define the scope of language-related requirements, and increase resources available to facilitate the work in this area.

Developer support
A core focus for the W3C is to support creation of specifications of web-

