Ensuring that the Web supports world wide users

Richard Ishida

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

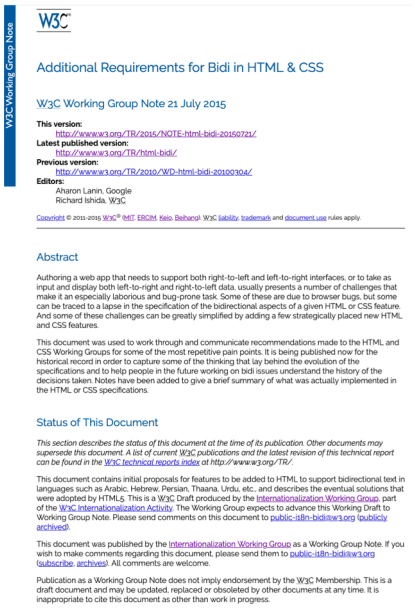
Build standards & applications that support a global Web

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/
Editors:
Aharon Lanin, Google
Richard Ishida, W3C

Copyright © 2011-2015 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 to a lapse in 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 in order to capture some of the thinking that lay behind the evolution of the specifications and to help people in the future working on bidi 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.

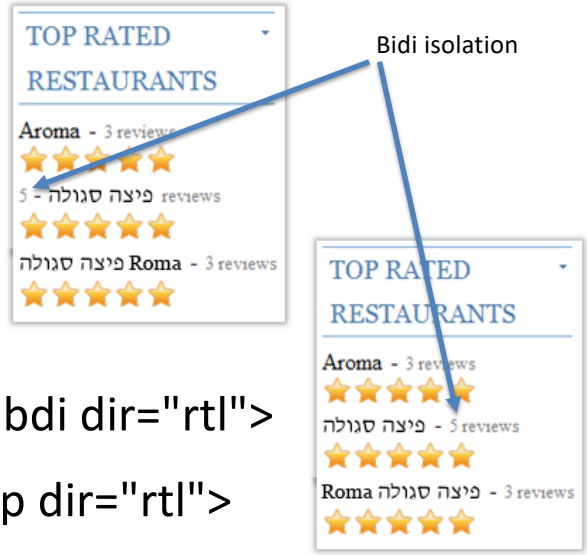
Status of This Document

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 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, Thaana, 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-i18n-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-i18n-bidi@w3.org (subscribe, archives). All comments are welcome.

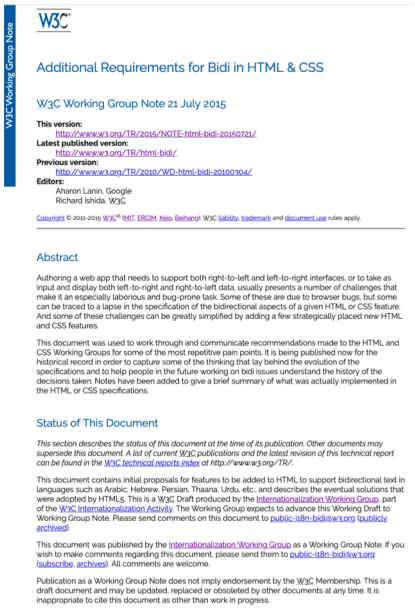
Publication as a Working Group Note does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.



<bdi dir="rtl">

<p dir="rtl">

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/
Editors:
Aharon Lanin, Google
Richard Ishida, W3C

Copyright © 2011-2015 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 to a lapse in 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 in order to capture some of the thinking that lay behind the evolution of the specifications and to help people in the future working on bidi 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.

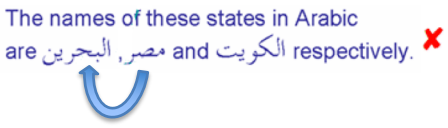
Status of This Document

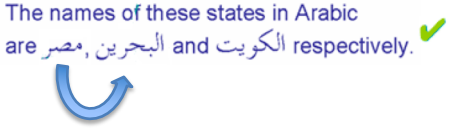
This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This document contains initial proposals for features to be added to HTML to support bidirectional text in languages such as Arabic, Hebrew, Persian, Thaana, 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-i18n-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-i18n-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.





string metadata

في HMTL5 يتم تحقيق ذلك بإضافة العنصر المضمن bdi.

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

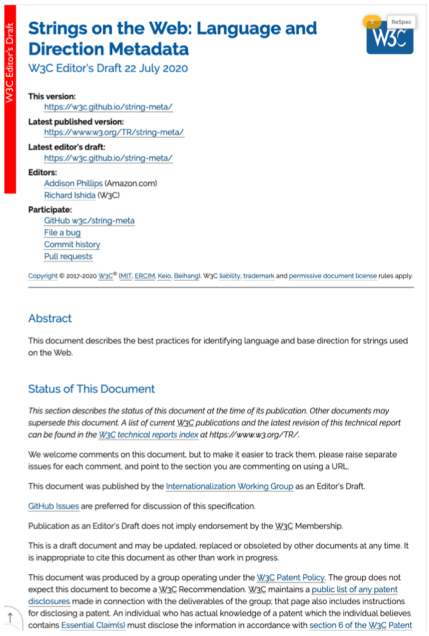
في HMTL5 يتم تحقيق ذلك بإضافة العنصر المضمن bdi.

string metadata

string metadata

string metadata

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/
Editors:
Addison Phillips (Amazon.com)
Richard Ishida (W3C)
Participate:
GitHub w3c/string-meta
File a bug
Commit history
Pull requests

Copyright © 2017-2020 W3C® (MIT, ERCIM, Keio, Beihang). 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

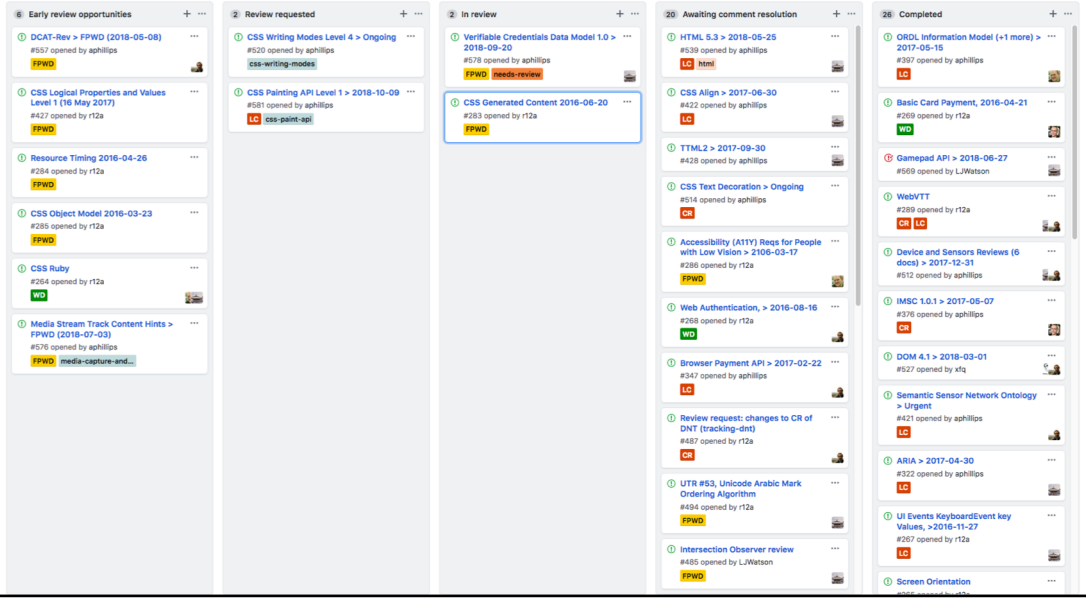
```
"id": "978-111887164-5",
"title": "HTML و CSS: تصميم و إنشاء مواقع الويب",
"authors": [ "Jon Duckett" ],
"language": "ar",
"pubDate": "2008-01-01",
"publisher": "مكتبة",
```

✗ تصميم و إنشاء مواقع الويب :CSS و HTML

✓ HTML و CSS: تصميم و إنشاء مواقع الويب

spec reviews

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



6 Early review opportunities
- DCAT-Rev > FPWD (2018-05-08) #557 opened by aphillips — FPWD
- CSS Logical Properties and Values Level 1 (16 May 2017) #427 opened by r12a — FPWD
- Resource Timing 2016-04-26 #284 opened by r12a — FPWD
- CSS Object Model 2016-03-23 #285 opened by r12a — FPWD
- CSS Ruby #264 opened by r12a — WD
- Media Stream Track Content Hints > FPWD (2018-07-03) #576 opened by aphillips — FPWD media-capture-and...

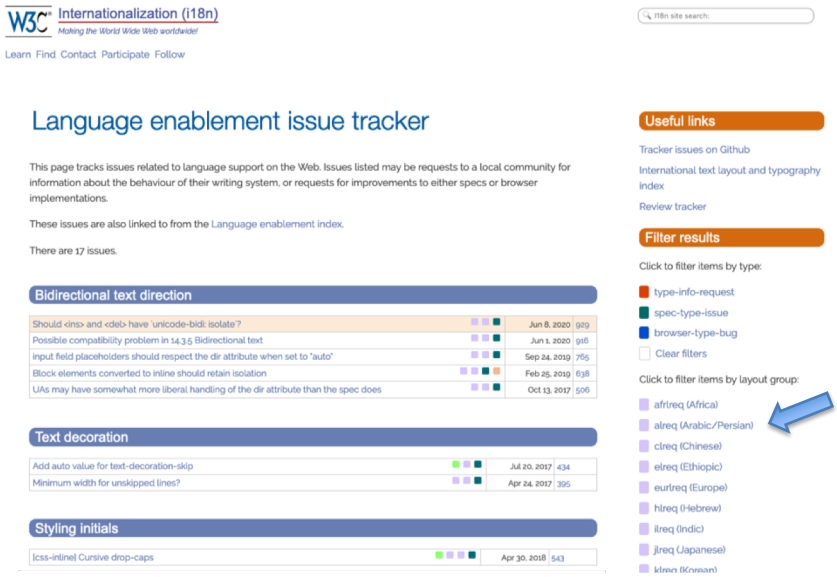
2 Review requested
- CSS Writing Modes Level 4 > Ongoing #520 opened by aphillips — css-writing-modes
- CSS Painting API Level 1 > 2018-10-09 #581 opened by aphillips — LC css-paint-api

2 In review
- Verifiable Credentials Data Model 1.0 > 2018-09-20 #578 opened by aphillips — FPWD needs-review
- CSS Generated Content 2016-06-20 #283 opened by r12a — FPWD

20 Awaiting comment resolution
- HTML 5.3 > 2018-05-25 #539 opened by aphillips — LC html
- CSS Align > 2017-06-30 #422 opened by aphillips — LC
- TTML2 > 2017-09-30 #428 opened by aphillips
- CSS Text Decoration > Ongoing #514 opened by aphillips — CR
- Accessibility (A11Y) Reqs for People with Low Vision > 2106-03-17 #286 opened by r12a — FPWD
- Web Authentication, > 2016-08-16 #268 opened by r12a — WD
- Browser Payment API > 2017-02-22 #347 opened by aphillips — LC
- Review request: changes to CR of DNT (tracking-dnt) #487 opened by r12a — CR
- UTR #53, Unicode Arabic Mark Ordering Algorithm #494 opened by r12a — FPWD
- Intersection Observer review #485 opened by LJWatson — FPWD

26 Completed
- ORDL Information Model (+1 more) > 2017-05-15 #397 opened by aphillips — LC
- Basic Card Payment, 2016-04-21 #269 opened by r12a — WD
- Gamepad API > 2018-06-27 #569 opened by LJWatson
- WebVTT #289 opened by r12a — CR LC
- Device and Sensors Reviews (6 docs) > 2017-12-31 #512 opened by aphillips
- IMSC 1.0.1 > 2017-05-07 #376 opened by aphillips — CR
- DOM 4.1 > 2018-03-01 #527 opened by xfq
- Semantic Sensor Network Ontology > Urgent #421 opened by aphillips — LC
- ARIA > 2017-04-30 #322 opened by aphillips — LC
- UI Events KeyboardEvent key Values, >2016-11-27 #267 opened by r12a — LC
- Screen Orientation

Current spec work for Arabic



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

Learn Find Contact Participate Follow

i18n site search:

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.

There are 17 issues.

Bidirectional text direction

Should <ins> and <del> have 'unicode-bidi: isolate'?	Jun 8, 2020	929
Possible compatibility problem in 14.3.5 Bidirectional text	Jun 1, 2020	916
input field placeholders should respect the dir attribute when set to "auto"	Sep 24, 2019	765
Block elements converted to inline should retain isolation	Feb 25, 2019	638
UAs may have somewhat more liberal handling of the dir attribute than the spec does	Oct 13, 2017	506

Text decoration

Add auto value for text-decoration-skip	Jul 20, 2017	434
Minimum width for unskipped lines?	Apr 24, 2017	395

Styling initials

[css-inline] Cursive drop-caps	Apr 30, 2018	543

Useful links

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

Filter results

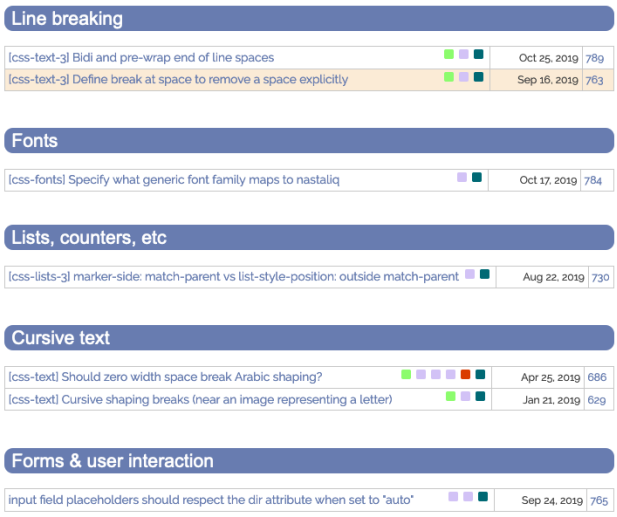
Click to filter items by type:

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

Click to filter items by layout group:

afrreq (Africa)
alreq (Arabic/Persian)
clreq (Chinese)
elreq (Ethiopic)
eurlreq (Europe)
hlreq (Hebrew)
ilreq (Indic)
jlreq (Japanese)
klreq (Korean)

Current spec work for Arabic



Line breaking

[css-text-3] Bidi and pre-wrap end of line spaces	Oct 25, 2019	789
[css-text-3] Define break at space to remove a space explicitly	Sep 16, 2019	763

Fonts

[css-fonts] Specify what generic font family maps to nastaliq	Oct 17, 2019	784

Lists, counters, etc

[css-lists-3] marker-side: match-parent vs list-style-position: outside match-parent	Aug 22, 2019	730

Cursive text

[css-text] Should zero width space break Arabic shaping?	Apr 25, 2019	686
[css-text] Cursive shaping breaks (near an image representing a letter)	Jan 21, 2019	629

Forms & user interaction

input field placeholders should respect the dir attribute when set to "auto"	Sep 24, 2019	765

'specdev' guidelines

W3C Editor's Draft

Internationalization Best Practices for Spec Developers

W3C Editor's Draft 07 August 2018

This version:
https://w3c.github.io/bp-i18n-specdev/
Latest published version:
https://www.w3.org/TR/international-specs/
Latest editor's draft:
https://w3c.github.io/bp-i18n-specdev/
Bug tracker:
File a bug (open bugs)
Editor:
Richard Ishida (W3C)
Github:
repository

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 provides advice to specification developers about how to incorporate requirements for international use. What is currently available here is expected to be useful immediately, but is still an early draft and the document is in flux, and will grow over time as knowledge applied in reviews and discussions can be crystallized into guidelines.

NOTE

Sending comments on this document

If you wish to make comments regarding this document, please raise them as github issues. Only send comments by email if you are unable to raise issues on github (see links below).

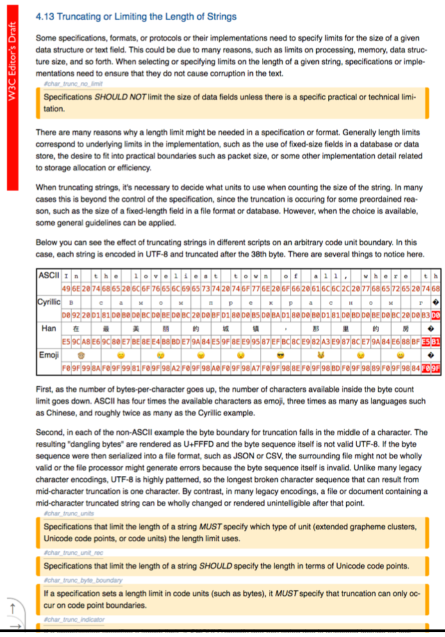
To make it easier to track comments, please raise separate issues or emails for each comment, and point to the section you are commenting on using a URL for the dated version of the document. All comments are welcome.

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

Comments regarding this document are welcome. Please send them to www-international@w3.org (archives).

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. 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.



W3C Editor's Draft

4.13 Truncating or Limiting the Length of Strings

Some specifications, formats, or protocols or their implementations need to specify limits for the size of a given data structure or text field. This could be due to many reasons, such as limits on processing, memory, data structure size, and so forth. When selecting or specifying limits on the length of a given string, specifications or implementations need to ensure that they do not cause corruption in the text.

Specifications *SHOULD NOT* limit the size of data fields unless there is a specific practical or technical limitation.

There are many reasons why a length limit might be needed in a specification or format. Generally length limits correspond to underlying limits in the implementation, such as the use of fixed-size fields in a database or data store, the desire to fit into practical boundaries such as packet size, or some other implementation detail related to storage allocation or efficiency.

When truncating strings, it's necessary to decide what units to use when counting the size of the string. In many cases this is beyond the control of the specification, since the truncation is occuring for some preordained reason, such as the size of a fixed-length field in a file format or database. However, when the choice is available, some general guidelines can be applied.

Below you can see the effect of truncating strings in different scripts on an arbitrary code unit boundary. In this case, each string is encoded in UTF-8 and truncated after the 38th byte. There are several things to notice here.

First, as the number of bytes-per-character goes up, the number of characters available inside the byte count limit goes down. ASCII has four times the available characters as emoji, three times as many as languages such as Chinese, and roughly twice as many as the Cyrillic example.

Second, in each of the non-ASCII example the byte boundary for truncation falls in the middle of a character. The resulting "dangling bytes" are rendered as U+FFFD and the byte sequence itself is not valid UTF-8. If the byte sequence were then serialized into a file format, such as JSON or CSV, the surrounding file might not be wholly valid or the file processor might generate errors because the byte sequence itself is invalid. Unlike many legacy character encodings, UTF-8 is highly patterned, so the longest broken character sequence that can result from mid-character truncation is one character. By contrast, in many legacy encodings, a file or document containing a mid-character truncated string can be wholly changed or rendered unintelligible after that point.

Specifications that limit the length of a string *MUST* specify which type of unit (extended grapheme clusters, Unicode code points, or code units) the length limit uses.

Specifications that limit the length of a string *SHOULD* specify the length in terms of Unicode code points.

If a specification sets a length limit in code units (such as bytes), it *MUST* specify that truncation can only occur on code point boundaries.

Developer support

Much work done to support RTL & bidi

Guidelines for developers

In-depth research and advice

Work still very much in progress

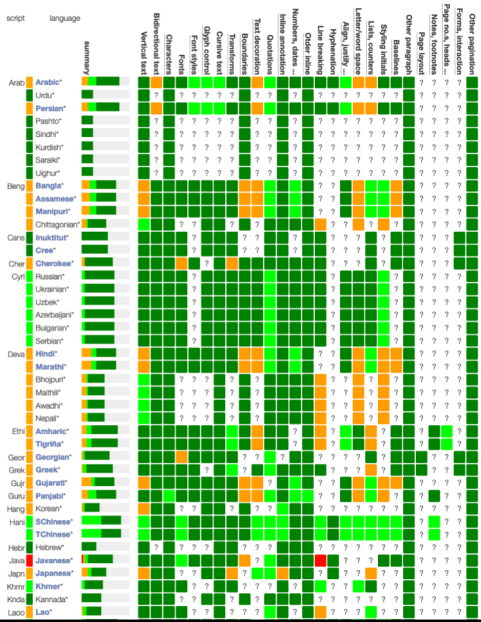
Language Enablement

Understand where the gaps are for users of the Web.

Muscat, Oman

language matrix

w3c.github.io/typography/gap-analysis/language-matrix



language matrix

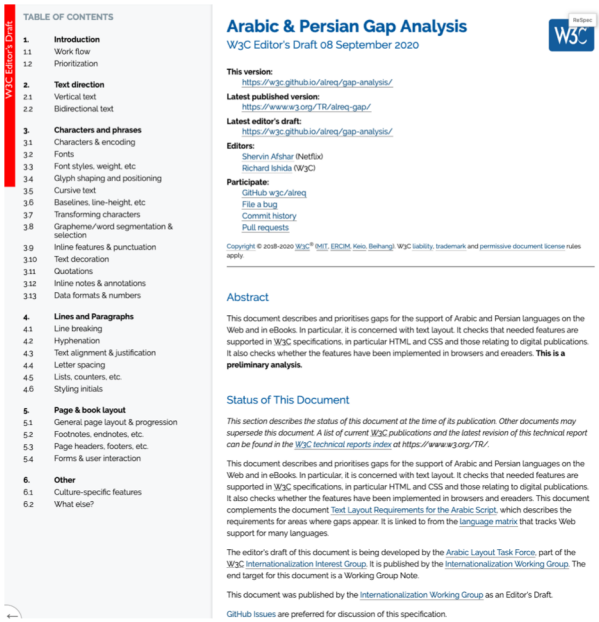


TABLE OF CONTENTS

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 Baselines, line-height, etc
3.7 Transforming characters
3.8 Grapheme/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?

W3C Editor's Draft

Arabic & Persian Gap Analysis

W3C Editor's Draft 08 September 2020

This version:
https://w3c.github.io/alreq/gap-analysis/
Latest published version:
https://www.w3.org/TR/alreq-gap/
Latest editor's draft:
https://w3c.github.io/alreq/gap-analysis/
Editors:
Shervin Afshar (Netflix)
Richard Ishida (W3C)
Participate:
GitHub w3c/alreq
File a bug
Commit history
Pull requests

Abstract

This document describes and prioritises gaps for the support of Arabic and Persian languages 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 Arabic and Persian languages 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 Text Layout Requirements for the Arabic Script, 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 Arabic Layout 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 an Editor's Draft.

GitHub Issues are preferred for discussion of this specification.

gap analysis

§ 3.5 Cursive text

If this script is cursive (eg. Arabic, N'Ko, Syriac, etc), are there problems or needed features related to the handling of cursive text? Do cursive links break if parts of a word are marked up or styled? Do Unicode joiner and non-joiner characters behave as expected? See available information or check for currently needed data

#222 Inline elements break cursive shaping

When elements surround part of a cursive run of text, and apply styling, the results often break the cursive joins. (See the results of trying to colour individual letters in the illustration below – successful on the left, unsuccessful on the right.)

ع ع ع

After some discussion, the CSS spec requires the following:

1. Markup alone around part of a joined up sequence must not disturb the joining behaviour.
2. Styling that doesn't affect the characters, such as text-decoration, must not break the joins.
3. Styling that does affect the shape of the characters should not break the joins, however the result is not well defined for complex glyph arrangements such as ligatures where the markup occurs between characters that make up the ligature.
4. Non-zero margins, padding, and borders, will break the join, as will isolation boundaries.

See the tests and results for a range of different styling effects.

Safari breaks cursive joining as soon as markup appears around a character, and so obviously fails for any type of styling application, too.

Firefox, Chrome, and Edge keep joins for styling that doesn't affect the shape of the characters (eg. text-decoration), and keeps it for colour changes, however they fail for changes in font-weight, font-style, and font-size, as well as for markup such as em and b tags.

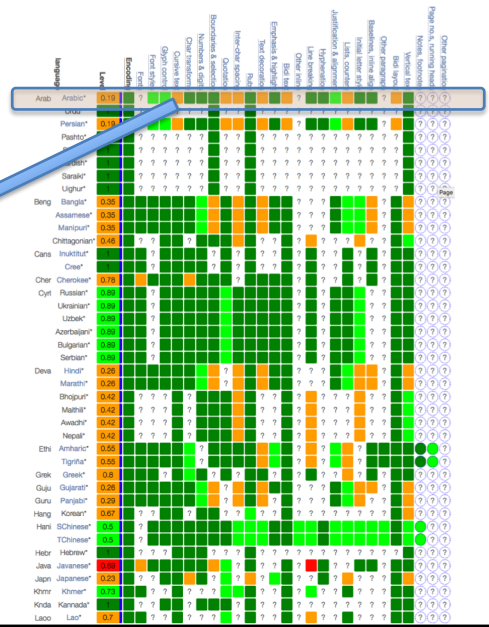
It would be useful to decide on the potential impact of the failures described here, so as to prioritise the issue. Is the inability to surround/style parts of a word a significant issue? It may be problematic when defining a term (using dfn) if the term is only part of the run of letters between spaces, eg. after the definite article.

#221 Text opacity shows glyph overlap

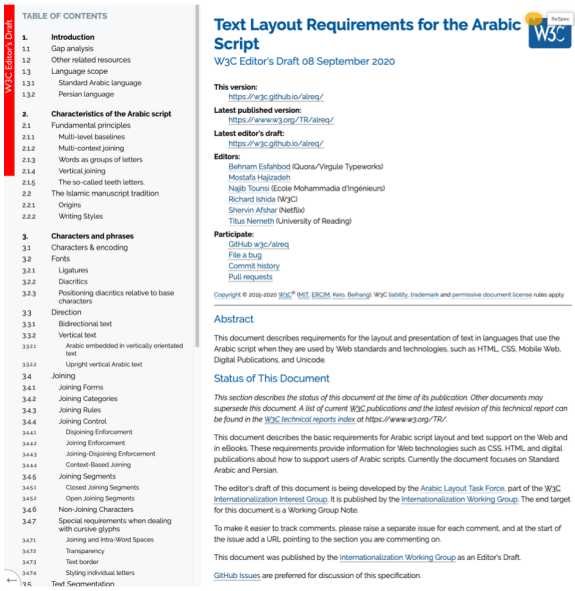
When opacity is applied to text, Firefox and Chrome produce dark patches where the cursive glyphs overlap, but Safari and Edge don't. See the test.

نشاط التدويل

The CSS Fonts spec was updated 2020-07-27 to clarify that opacity should be applied to a



language enablement requirements



W3C Editor's Draft

TABLE OF CONTENTS

Text Layout Requirements for the Arabic Script

W3C Editor's Draft 08 September 2020

This version:
https://w3c.github.io/alreq/

Latest published version:
https://www.w3.org/TR/alreq/

Latest editor's draft:
https://w3c.github.io/alreq/

Editors:
Behnam Esfahbod (Quora/Virgule Typeworks)
Mostafa Hajizadeh
Najib Tounsi (Ecole Mohammadia d'Ingénieurs)
Richard Ishida (W3C)
Shervin Afshar (Netflix)
Titus Nemeth (University of Reading)

Participate:
GitHub w3c/alreq
File a bug
Commit history
Pull requests

Abstract

This document describes requirements for the layout and presentation of text in languages that use the Arabic 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 document describes the basic requirements for Arabic script layout and text support on the Web and in eBooks. These requirements provide information for Web technologies such as CSS, HTML and digital publications about how to support users of Arabic scripts. Currently the document focuses on Standard Arabic and Persian.

The editor's draft of this document is being developed by the Arabic Layout 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.

To make it easier to track comments, please raise a separate issue for each comment, and at the start of the issue add a URL pointing to the section you are commenting on.

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

GitHub Issues are preferred for discussion of this specification.

language enablement requirements

§ *3.4.7.2 Transparency*

Arabic fonts achieve joining by overlapping letters. A left-joining letter extends out of its bounding box from the left side and a right-joining letter extends out of its bounding box from the right side. Making each letter transparent can expose these overlapping joinings, which should be avoided. Joining the paths of the joined letter into a single shape can remove the overlappings and create the good results.

Figure 35: Applying transparency to Arabic letters should not expose their joining overlaps.

§ *3.4.7.3 Text border*

When adding text border, simply adding a border to each letter shape fails to produce the proper result for the Arabic script. A joined letter should not be separated from its joined neighbors by adding border. Like transparency, a way to avoid this is to unify glyph paths into a single big path for all the letters that are joined and add border around that path.

Figure 36: Text border should not expose joinings.

§ *3.4.7.4 Styling individual letters*

For educational, technical, or even aesthetic reasons, users might want to apply a specific style to a single letter (or a few letters) in a word. For example, Figure 37 is the logo of the largest telecommunications provider in Oman.

Figure 37: Colour changes across joining characters in the logo for Omantel.

This should not break the letter's joining with its neighbors, as shown in Figure 38.

Figure 38: Applying style to a single letter should not interfere with its joining properties.

text layout index

w3c.github.io/typography/

TABLE OF CONTENTS

1. Introduction
2. Text direction
2.1 Vertical text
2.2 Bidirectional text direction
3. Characters & phrases
3.1 Fonts
3.2 Font styles, weight, etc.
3.3 Glyph shaping & positioning
3.4 Cursive text
3.5 Transforming characters
3.6 Text segmentation & selection
3.7 Punctuation
3.8 Text decoration
3.9 Quotations
3.10 Inline notes & annotations
3.11 Numbers & digits
3.12 Other inline features
4. Lines & paragraphs
4.1 Line breaking
4.2 Hyphenation
4.3 Text alignment & justification
4.4 Word & letter spacing
4.5 Lists, counters, etc
4.6 Styling initials
4.7 Baselines & inline alignment
5. Layout & pages
5.1 General page layout and progression
5.2 Grids & tables
5.3 Notes, footnotes, etc
5.4 Page headers, footers, etc
5.5 Forms & user interaction
6. 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

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

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.

4.2 Hyphenation §

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

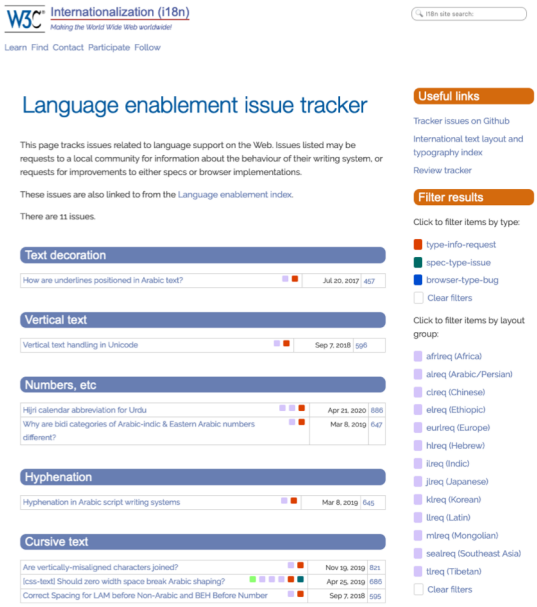
GitHub resources
- All issues
- Requests for information
- Spec issues
- Useful discussions
- Type samples

Spec links
- *CSS3 Text*: Hyphenation: the hyphens property

Tests
- CSS Text: Hyphens

Gap analysis
- Dutch
- Hungarian
- Javanese

questions about Arabic



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.

There are 11 issues.

Text decoration
- How are underlines positioned in Arabic text? — Jul 20, 2017 | 457

Vertical text
- Vertical text handling in Unicode — Sep 7, 2018 | 596

Numbers, etc
- Hijri calendar abbreviation for Urdu — Apr 21, 2020 | 686
- 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

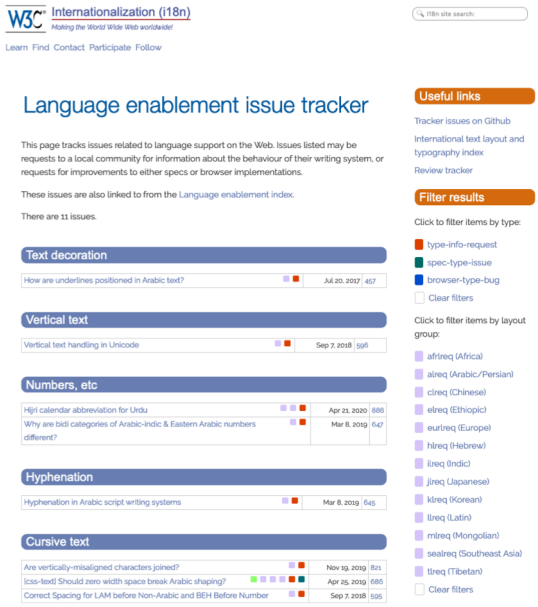
Cursive text
- Are vertically-misaligned characters joined? — Nov 19, 2019 | 821
- [css-text] Should zero width space break Arabic shaping? — Apr 25, 2019 | 686
- Correct Spacing for LAM before Non-Arabic and BEH Before Number — Sep 7, 2018 | 595

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

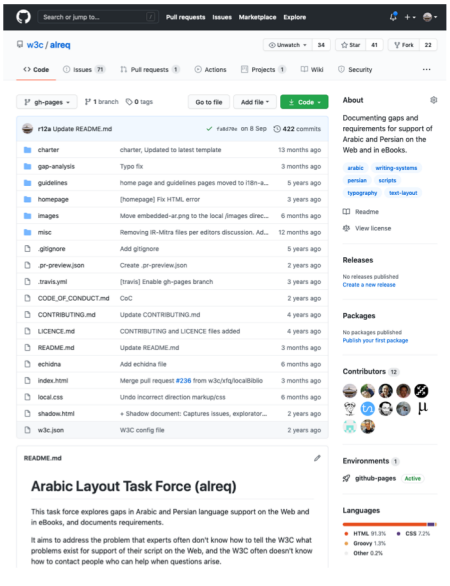
Click to filter items by layout group:
afrlreq (Africa)
alreq (Arabic/Persian)
clreq (Chinese)
elreq (Ethiopic)
eurlreq (Europe)
hlreq (Hebrew)
ilreq (Indic)
jlreq (Japanese)
klreq (Korean)
llreq (Latin)
mlreq (Mongolian)
sealreq (Southeast Asia)
tlreq (Tibetan)
Clear filters

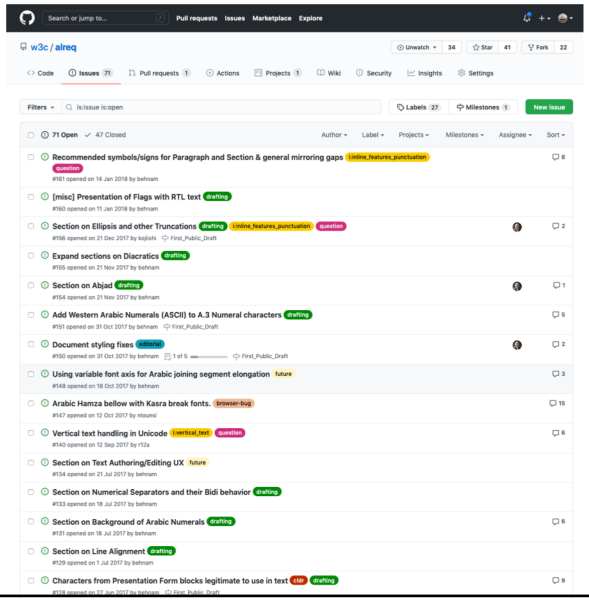
questions about Arabic



networks

w3c.github.io/sealreq/





Language Enablement

Language matrix

Gap-analysis

Layout requirements

Expert networks

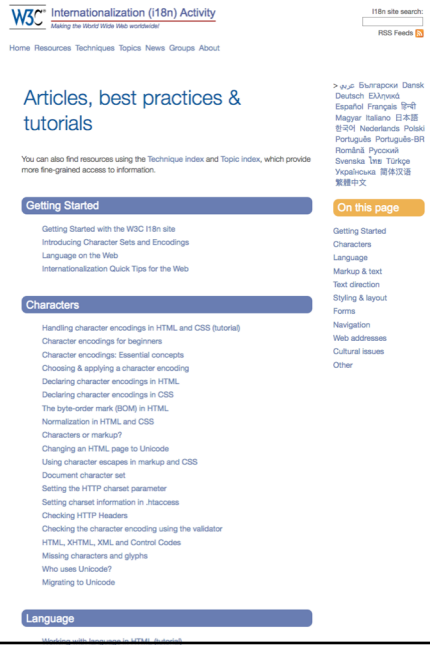
Need more experts!

Author support

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

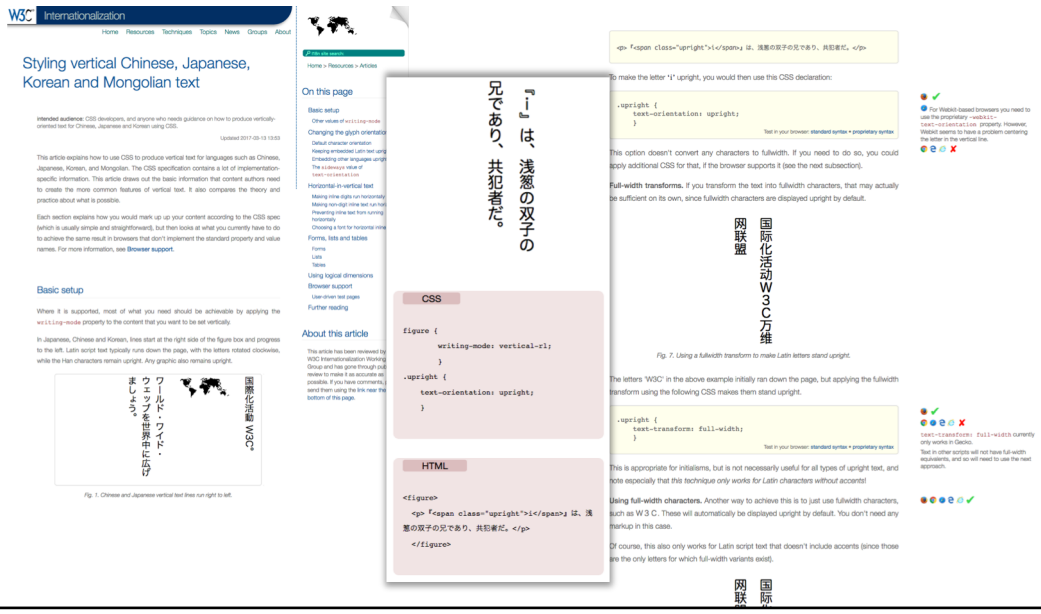
Wakan, Oman

articles



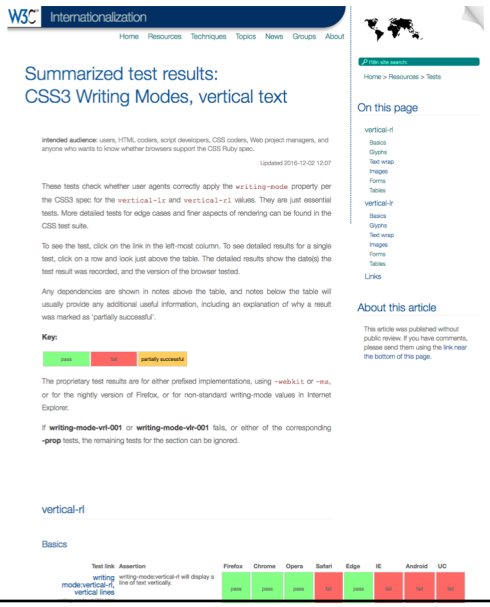
- Getting started
- Characters
- Language
- Markup & text
- Text direction
- Styling & layout
- Forms
- Navigation
- Web addresses
- Cultural issues

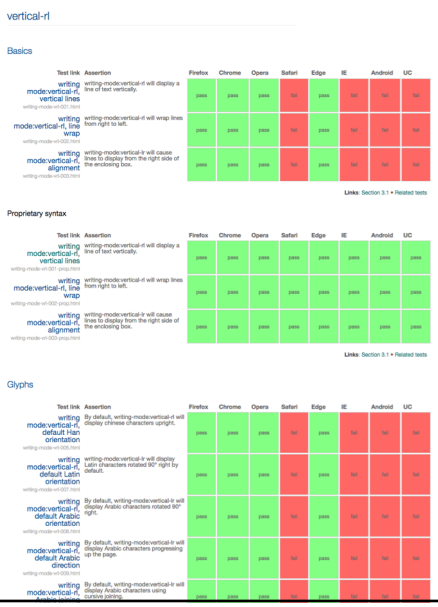
vertical text guidelines



i18n test suite

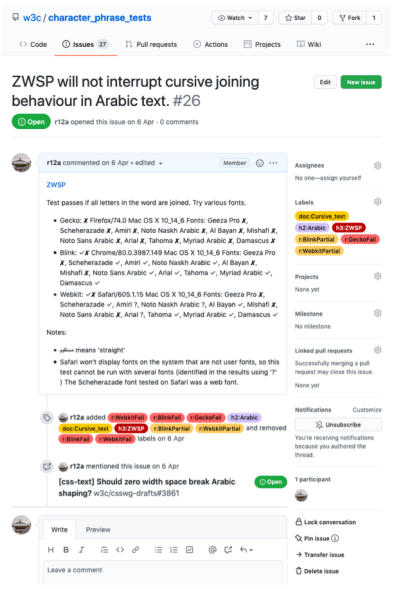
w3.org/international/tests





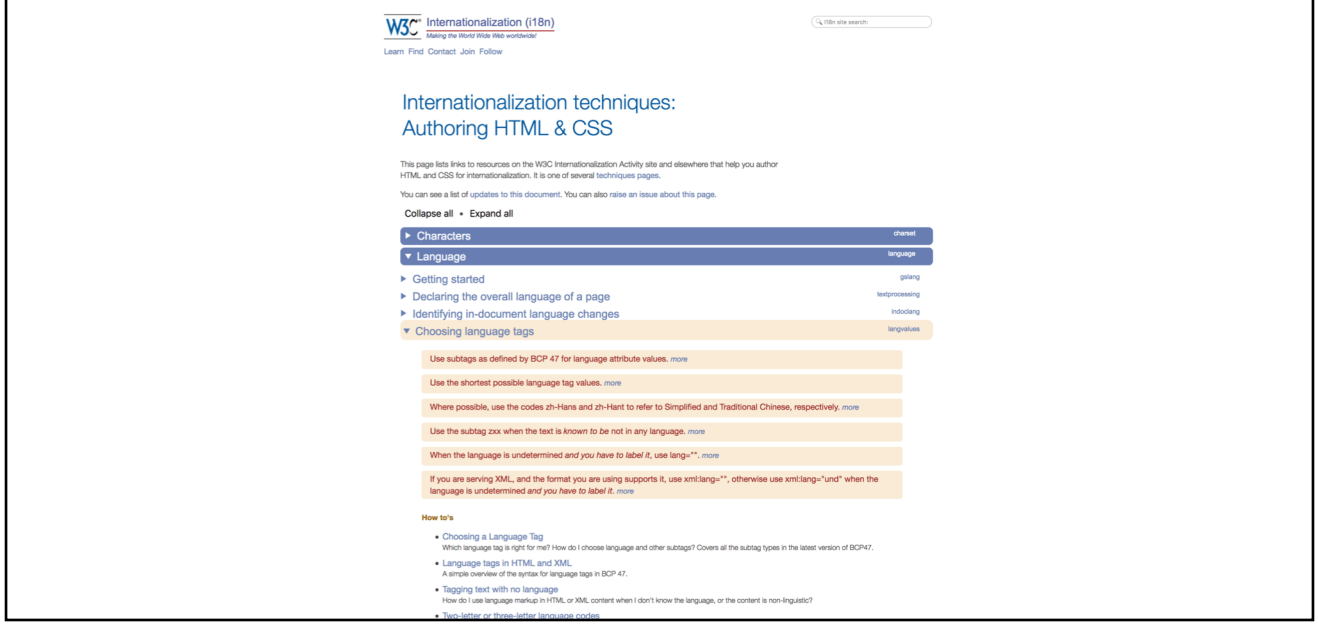
interactive tests

w3.org/international/tests



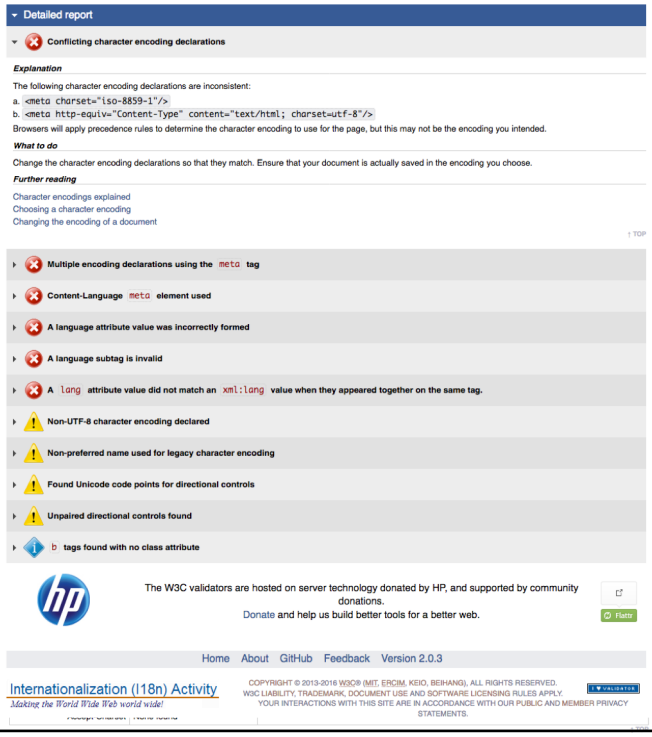
techniques index

w3.org/International/techniques/authoring-html



i18n checker

validator.w3.org/i18n-checker



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 Arabic 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 recent decades, but as Web penetration in language communities 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 stakeholders all around the world, there must be a collaboration of language experts, Web 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 organizations and experts who care about these problems 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.
- **Developer support** appeals most directly to tech companies that are building the infrastructure for a global Web and supporting W3C standards groups.
- **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 these goals 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 identified and addressed. Text layout is an area of particular interest, and is concerned with things such as rules for line-breaking & justification, local approaches to expressing emphasis or decorating text, localizing counter styles, supporting bidirectional text in markup, initial-letter styling, hyphenation, page layout, and so on. These typographic conventions are often very different from the Western norm in languages that use writing systems such as Arabic, Devanagari, Thai, Mongolian, and so forth.

In Web pages and in digital publishing, needs can be addressed by improving W3C standards for rendering text (such as CSS, WebVTT, SVG, etc.), and markup (such as HTML). The goal here is to ensure that the Web supports the native typographic features that users around the world are used to, and enables users to interact with the Web in line with long-standing print traditions. The language matrix captures an overview of where work is needed.

To achieve this goal, the W3C needs to assess current support for the world's languages on the Web, identify gaps, prioritize them, develop requirements, and then take steps to close the gaps. To do so, it needs to establish a network of experts who can advise on 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 creators of specifications, of system-

Bled, Slovenia

www.w3.org/International

ishida@w3.org