

Internationalization

Content authoring that supports worldwide use of the Web

Richard Ishida
W3C

Objectives

- ▶ Discuss various aspects of content authoring that support universal access to the Web
- ▶ You will understand
 - ▶ how to declare and use characters & character encodings in X/HTML & CSS
 - ▶ how to declare the language of the document or parts of the document
 - ▶ how to use Chinese or other scripts in URIs according to the latest standards
- ▶ You will see some of the new CSS style capabilities that may soon be available for Asian languages

2

W3C

Outline

- Character sets & encoding
- Identifying language
- IDN & IRIs
- East Asian typography

4

W3C

Outline

- **Character sets & encoding**
 - Character set vs. character encoding
 - The Document Character Set
 - Choosing an encoding
 - Serving HTML & XHTML
 - Declaring the document encoding
 - Entities & Numeric Character References
 - Care & feeding of characters
- Identifying language
- IDN & IRIs
- East Asian typography

5

W3C

Character set vs. character encoding

6



▶ An important initial distinction

7

Character set

The set of atomic text elements you will use for a particular purpose.

Character Encoding

The way these abstract characters are mapped to numbers for manipulation in a computer.



Character set vs character encoding

▶ A single character set, such as Unicode, may have more than one character encoding.

9

	A	κ	好	不
Code point	U+0041	U+05D0	U+597D	U+233B4
UTF-8	41	D7 90	E5 A5 BD	F0 A3 8E B4
UTF-16	00 41	05 D0	59 7D	D8 4C DF B4
UTF-32	00 00 00 41	00 00 05 D0	00 00 59 7D	00 02 33 B4



The Document Character Set

10



▶ What is the Document Character Set?

11

- the logical model that describes how XML and HTML are processed
- for XML and HTML (from version 4.0): the Universal Character Set (UCS) defined by both ISO/IEC 10646 and Unicode standards
- it does not mean that all HTML and XML documents have to be encoded as Unicode !



▶ What is the Document Character Set?

12

- means that documents can only contain characters defined by Unicode
- any encoding can be used for your document as long as it is properly declared and a subset of the Unicode repertoire
- values of numeric character references (such as `ǵ` and `ǵ` for `ğ`) are interpreted as Unicode characters - no matter what encoding you use for your document

See GEO FAQ: Document character set

<http://www.w3.org/International/questions/qa-doc-charset.html>



Choosing an encoding

13



▶ Consider using Unicode

14

- supports many languages, enabling the use of a single encoding across all pages and forms, regardless of language
- eliminates the need for server-side logic to determine the character encoding for each page served or each incoming form submission
- allows many more languages to be mixed on a single page than almost any other choice

Although there are other multi-script approaches (such as ISO-2022 and GB18030), Unicode generally provides the best combination of user agent and script support.



▶ If you don't use Unicode

15

- select an encoding that maximizes the opportunity to directly represent characters / minimizes the need to represent characters by character escapes
- select commonly supported encodings & check that user agents adequately support the encoding selected
- consider a solution that minimizes complexity when dealing with multiple languages and scripts
- note that support for a given encoding does not necessarily imply support for all writing systems that that encoding supports



Serving HTML & XHTML

16



▶ XHTML & MIME types

17

HTML	text/html	
XHTML	text/html	Use the compatibility guidelines in XHTML Spec, Appendix C !
	application/xhtml+xml	
	application/xml	
	text/xml	



▶ XHTML & MIME types

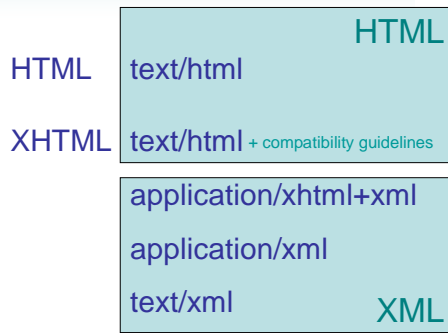
18

HTML	text/html	
XHTML	text/html	Use the compatibility guidelines in XHTML Spec, Appendix C !
	application/xhtml+xml	
	application/xml	
	text/xml	



▶ XHTML & MIME types

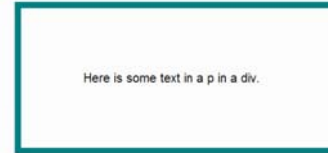
19



▶ 'Standards' vs 'Quirks' modes

20

Test file for Standards Mode



Here is some text... in a p tag
Here is some ... that's not.



▶ 'Standards' vs 'Quirks' modes

21

Test file for Standards Mode



Here is some text... in a p tag
Here is some ... that's not.



▶ 'Standards' vs 'Quirks' modes

24

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>xhtml document</title>
  <style type="text/css">
    body { background: white; color: black; font-family: arial, sans-serif; font-size: 40px; }
    p { font-size: 50%; }
    h1 { font-size: 24px; }
  </style>
</head>
<body>
  <h1>Test file for Standards Mode</h1>
  <div style="margin: 50px; width: 300px; padding: 100px; border: 10px solid teal;">
    <p>Here is some text in a p in a div. </p>
  </div>
  <table border="1">
    <tr><td>Here is some text...</td>
    <td>...in a p tag</td>
    </tr>
    <tr><td>Here is some ...</td>
    <td>... that's not.</td>
    </tr>
  </table>
</body>
</html>
```



'Standards' vs 'Quirks' modes

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>xhtml document</title>
  <style type="text/css">
  body { background: white; color: black; font-family: arial, sans-serif; font-size: 40px; }
  p { font-size: 50%; }
  h1 { font-size: 24px; }
  </style>
</head>
<body>
  <h1>Test file for Standards Mode</h1>
  <div style="margin: 50px; width: 300px; padding: 100px; border: 10px solid teal;">
  <p> Here is some text in a p in a div. </p>
  </div>
  <table border="1">
  <tr><td><p>Here is some text...</p></td>
  <td><p>...in a p tag</p></td>
  </tr>
  <tr><td>Here is some ...</td>
  <td>... that's not.</td>
  </tr>
  </table>
</body>
</html>
```

25



Summary of assumptions & recommendations

- we assume standards mode and relatively up to date user agents
- use XHTML where possible
- XHTML served as xml is still not widely supported
- XHTML served as XML should be served as application/xhtml+xml
- we assume that some people will not want to use the XML declaration ie. <?xml version="1.0" encoding="utf-8"?>

26



Declaring the document encoding

27



Basic scenarios

- HTTP <?xml .. <meta ..
- HTML
- XHTML (text/html)
- XHTML (XML)

28



▶ Where appropriate, declare the page's character encoding by setting the charset parameter in the HTTP Content-Type header.

29

- method depends on the server
- server may have default settings
- users may be able to override default settings

Eg .htaccess files in Apache

- AddType 'text/html; charset=UTF-8' html
- <Files ~ "events\.html">
ForceType 'text/html; charset=UTF-8'
</Files>



▶ Where appropriate, declare the page's character encoding by setting the charset parameter in the HTTP Content-Type header.

30

- + • user agents can easily find the information
- highest priority in case of conflict, so should be used where transcoding done by the server
- • more difficult for content authors to change the setting - especially when dealing with an ISP
- server settings may get out of synch with the document
- doesn't cater for documents read from CD or hard disk
- may not facilitate processing, eg XSLT or translation



▶ For XHTML served as text/html, where practical use an XML declaration with an encoding attribute.

31

```
<?xml version="1.0" encoding="UTF-8"?>
```

Character set name

- not required for UTF-8 or UTF-16, but useful anyway
- find a name at <http://www.iana.org/assignments/character-sets>
- use the preferred name when aliases are available
- avoid using unregistered names (ie. x-...)



▶ For XHTML served as text/html, where practical use an XML declaration with an encoding attribute.

32

- + • useful when editing or processing the file as XML, eg. using XSLT
- helps developers, testers, or translation production managers who want to perform a visual check of a document
- allows the document to be read correctly when not on the server
- the XHTML spec says you should
- • knocks Internet Explorer documents into Quirks mode
- not actually needed for HTML documents



▶ For XHTML served as `application/xhtml+xml`, always use an XML declaration with an encoding attribute.

33

- +
 - useful when processing the file as XML, eg. using XSLT
 - developers, testers, or translation production managers may want to perform a visual check of a document
 - the XHTML spec says you should
 - there is likely to be no other in-document alternative



▶ For HTML documents and XHTML documents served as `text/html`, always use the `<meta>` element to explicitly declare the document's character encoding.

34

```
<meta http-equiv="Content-type"
      content="text/html; charset=UTF-8" />
```

Character set name
↙

- required for all encodings, including UTF-8 or UTF-16
- put as near as possible to the top of the file (ie. before `<title>`)
- find a name at <http://www.iana.org/assignments/character-sets>
- use the preferred name when aliases are available
- avoid using unregistered names (ie. `x-...`)



▶ For HTML documents and XHTML documents served as `text/html`, always use the `<meta>` element to explicitly declare the document's character encoding.

35

- +
 - enables the document to be edited or read from CD or hard disk, etc.
 - helps developers, testers, or translation production managers who want to perform a visual check of a document
 - the XHTML spec says you should
 - there is likely to be no other in-document alternative



▶ Basic scenarios

36

	HTTP	<code><?xml ..</code>	<code><meta ..</code>
HTML	(✓)	X	✓
XHTML (<code>text/html</code>)	(✓)	(✓)	✓
XHTML (<code>XML</code>)	(✓)	✓	X



▶ Declare encoding for your CSS style sheets too

```
@charset "utf-8";
```

- ◆ must be the first thing in the file
- ◆ beware of the UTF-8 signature/BOM
- ◆ only necessary for external, linked style sheets
- ◆ likely to become increasingly important in the future
- ◆ particularly important if:
 - ◆ your style sheet contains non-ASCII values for the content property, or
 - ◆ refers to non-ASCII element or attribute names or values

37



▶ Precedence rules for XHTML/HTML encoding

1. HTTP Content-Type
2. XML declaration
3. meta statement
4. link charset attribute

- ◆ HTTP as a priority supports transcoding
- ◆ Use the link charset attribute with care

38



▶ Precedence rules for CSS

1. HTTP Content-Type
2. @charset rule
3. <link charset=".." rel="stylesheet" ... />

- ◆ HTTP as a priority supports transcoding
- ◆ Use the link charset attribute with care

39



Entities and Numeric Character References (NCRs)

40



- ▶ Only use escapes for characters in exceptional circumstances

á

- `á` hex NCR
- `á` decimal NCR
- `á` character entity
- `\E1` CSS escape

NCR = Numeric Character Reference

41



- ▶ Only use escapes for characters in exceptional circumstances
- ▶ Create pages using an encoding that supports all the characters you need

Jako efektivnější se nám jeví pořádání tzv. Road Show prostřednictvím našich autorizovaných dealerů v Čechách a na Moravě, které proběhnou v průběhu září a října.

42



- ▶ Only use escapes for characters in exceptional circumstances
- ▶ Create pages using an encoding that supports all the characters you need

Jako efektivnější se nám jeví pořádání tzv. Road Show prostřednictvím našich autorizovaných dealerů v Čechách a na Moravě, které proběhnou v průběhu září a října.

43



- ▶ Only use escapes for characters in exceptional circumstances

- syntax-related characters
 - < (<) > (>) & (&) " (")
- characters not supported by the document encoding
 - it may be better to change the encoding!
- characters not supported by the input tools
- characters that are invisible or ambiguous
 - eg. `&rim;` / `‏`
 - eg. ` ` / ` `

44



- ▶ Also bear in mind...
- NCRs always reference the Unicode code point, no matter what encoding you used !
 - suggest you use Hex values rather than Decimal – easier to look up
 - character entities can cause problems for processing as XML
 - use a single value for supplementary characters, not two
 ie. `𒍅` not `��`



Care and feeding of characters



▶ Some Unicode characters are not suitable for use with markup

Names/ Description	Short Comment
Line and paragraph separator	use <code><xhtml:br /></code> , <code><xhtml:p></xhtml:p></code> , or equivalent
BIDI embedding controls (LRE, RLE, LRO, RLO, PDF)	Strongly discouraged in [HTML 4.0]
Activate/Inhibit Symmetric swapping	Deprecated in Unicode
Activate/Inhibit Arabic form shaping	Deprecated in Unicode
Activate/Inhibit National digit shapes	Deprecated in Unicode
Interlinear annotation characters	Use ruby markup
Byte order mark / ZWNBSP	Use only as byte order mark. Use U+2060 Word Joiner instead of using U+FEFF as ZWNBSP
Object replacement character	Use markup, e.g. HTML <code><object></code> or HTML <code></code>
Scoping for Musical Notation	Use an appropriate markup language
Language Tag codepoints	Use <code>xhtml:lang</code> and/or <code>xml:lang</code>

Unicode in XML & Other Markup Languages
<http://www.w3.org/TR/unicode-xml/>



▶ Other Unicode characters are OK

Names/ Description	Short Comment
Various	No-break space, Soft Hyphen, Combining Grapheme Joiner, Non breaking Hyphen, Word Joiner, etc.
Zero-width Joiners (ZWJ and ZWNJ)	eg. required for Persian
Implicit directional marks (LRM and RLM)	
Subtending marks	common feature in the Arabic and Syriac scripts
Variation Selectors	eg. required for Mongolian
Ideographic Description Characters	indicate the composition of ideographs
Etc...	

Unicode in XML & Other Markup Languages
<http://www.w3.org/TR/unicode-xml/>



'Compatibility characters' vary in appropriateness

49

Names/ Description	Examples	Verdict
Circled letters and digits used for list item markers	⓪ Ⓛ Ⓜ Ⓝ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ	OK
Parenthesized or dotted number used as list item marker	(1) (2) (3)	use list item marker style
Arabic Presentation forms	ع ع ف ح	normalize
Half-width and full-width characters	〒 1 3 7 a b c d	OK
Superscripted and subscripted characters	^{1 2 3} _{1 2 3}	use <sup> markup
Etc...		

moral – use markup where possible

Unicode in XML & Other Markup Languages
<http://www.w3.org/TR/unicode-xml/>



Normalization

50

NFC Ízelítőül
 NFD Ízeliótoóüöí

Ha a világ beszélni akarna, Unicode-ul szólna meg. Regisztráljon már most a Tizedik Nemzetközi Unicode Konferenciára, melyet 1997. március 10-12-én rendeznek Mainz-ban, Németországban. Ezen a konferencián az iparág több neves szakértője is részt vesz. Ízelítőül a témakból: a világháló és a Unicode nemzetköziesítése és lokalizálása, a Unicode alkalmazása működő rendszerekben és alkalmazásokban, szövegelrendezésnél, és többnyelvű számítógépeken.



When creating Unicode text that has the potential for normalization, use NFC

51

- Normalization Form C
- closest to actual practice
- compact
- the Character Model for the World Wide Web recommends that text on the Web be in NFC form facilitates clarity and checking



Character sets & encoding

52

- Character sets & encoding
 - The Document Character Set
 - Choosing an encoding
 - Serving HTML & XHTML
 - Declaring the document encoding
 - Entities & Numeric Character References
 - Care & feeding of characters
- Identifying language
- IDN & IRIs
- East Asian typography



- Character sets & encoding
- Identifying language
 - Declaring the language of text
 - Language attribute values
 - Negotiating language with the server
 - Styling using the language attribute
- IDN & IRIs
- East Asian typography

53



Declaring the language of text

54



▶ Always declare the language of the document as a whole in the <html> tag

- ◆ HTML: use the lang attribute in the <html> tag


```
<html lang="zh-CN">
```
- ◆ XHTML as text/html: use both the lang attribute and the xml:lang attribute in the <html> tag


```
<html lang="zh-CN" xml:lang="zh-CN"
  xmlns="http://www.w3.org/1999/xhtml">
```
- ◆ XHTML as XML: use the xml:lang attribute in the <html> tag


```
<html xml:lang="zh-CN"
  xmlns="http://www.w3.org/1999/xhtml">
```

55



▶ Always declare changes in the language of the text

- ◆ HTML: use the lang attribute


```
<p>The French for <em>Cat</em> is
  <em lang="fr">chat</em>.</p>
```
- ◆ XHTML as text/html: use both the lang attribute and the xml:lang attribute (xml:lang takes precedence)


```
<p>The title in Chinese is <span lang="zh-CN"
  xml:lang="zh-CN">中国科学院文献情报中心</span>.</p>
```
- ◆ use a span element if there is nothing to hang the information on

56



- ▶ Why declare the language?
 - extremely important for screen readers and accessibility
 - allows for stylistic variations
 - some browsers use language to determine appropriate fonts for Simplified vs Traditional Chinese vs Japanese
 - allows for extraction of language-specific elements, eg. using XSLT's lang() function in XPath
 - etc.

57



Language attribute values

58



- ▶ Use RFC 3066 rules
 - HTML 4.01 specification still says RFC 1766
 - RFC 3066 replaced RFC 1766, so use this !
- `<html lang="en-GB">`
- language optional subcode(s)
country, dialect, etc.

59





- ▶ Use RFC 3066 rules
 - `<html lang="en-GB">`
 - case insensitive, A-Z, a-z, 0-9, up to eight letters
 - usually written lowercase, by convention
 - all 2-letter and 3-letter codes must be ISO 639 codes
 - don't use 3-letter ISO 639 codes if 2-letter code exists

60



▶ Use RFC 3066 rules




61

- `<html lang="en-GB">`  i... reserved for IANA-defined registrations
- language  x... user-defined codes, eg. x-ishidic

W3C

▶ Use RFC 3066 rules

62

- `<html lang="en-GB">`  optional
- optional sublag(s) 
- dialect, country, etc. 
- case insensitive, A-Z, a-z, 0-9, two to eight letters
 - usually written uppercase, by convention
 - 2-letter codes must be ISO 3166 codes
 - 3- to 8-letter codes can be registered with IANA
 - more than 2 sublags possible, eg. en-UK-geordie – no special rules apply

W3C

▶ Use IANA assigned language tags

63

- IANA defines codes for specific combinations, eg. zh-guoyu, zh-hakka, zh-min, zh-min-nan, zh-wuu, ...
- includes codes for Simplified vs Traditional Chinese

```
<p lang="zh-Hans" xml:lang="zh-Hans">
  当世界需要沟通时，请用Unicode！</p>
```

```
<html lang="zh-Hant" xml:lang="zh-Hant">
  當世界需要溝通時，請用統一碼 (Unicode) </p>
```

- register your needed codes with IANA !

W3C

▶ Other points about language tags

64

- can be applied to other media (eg. audio objects) not just text
- used at the beginning of a document, it identifies the intended audience, rather than all the languages in the document
- 'en-GB' should also match 'en' – although this is not always the case for all implementations (eg. Apache language negotiation)
- xml now provides a means to prevent inheritance of language using the null string, ie.

```
xml:lang=""
```

W3C

▶ Issues with language tags

- doesn't cover all needs, eg. generic Latin-American Spanish
- some lack of clarity between use for language vs locale designation
- many more language codes needed for the approximately 6,000 world languages
- people working on improvements include ISO TC37, SIL, W3C, etc.
- register tags you need with IANA

65



Negotiating language with the server

66



▶ The server may be set up to serve alternative versions of a document in a particular language based on HTTP information sent by the user agent

```
GET /Press/1998/CSS2-REC HTTP/1.1
Host: www.w3.org
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.5a) Gecko/20030728 Mozilla Firebird/0.6.1
Accept:
text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
```

Accept-Language: en-GB,en;q=0.7,chs-Hans;q=0.3

```
Accept-Encoding: gzip,deflate
Accept-Charset: UTF-8,*
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.w3.org/International/questions/qa-lang-priorities.html
Cookie: absence_stuff=multi&0&group_id&2&chunks&size&6
If-Modified-Since: Tue, 12 May 1998 22:18:49 GMT
If-None-Match: "3558cac936f99e2b"
Cache-Control: max-age=0
```

REQUEST

REPLY

```
HTTP/1.1 200 OK
Date: Wed, 05 Nov 2003 10:46:04 GMT
Server: Apache/1.3.28 (Unix) PHP/4.2.3
Content-Location: CSS2-REC.en.html
Vary: negotiate,accept-language,accept-charset
TCN: choice
P3P:
policyref=http://www.w3.org/2001/05/P3P/p3p.xml
Cache-Control: max-age=21600
Expires: Wed, 05 Nov 2003 16:46:04 GMT
Last-Modified: Tue, 12 May 1998 22:18:49 GMT
ETag: "3558cac936f99e2b"
Accept-Ranges: bytes
Content-Length: 10734
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Content-Language: en

67



▶ Exercise caution in acting upon language information sent by the user agent

- the user may have never checked the Accept-Language setting
- user agent may send a request that specifies only a language and not a region (a particular issue if trying to establish locale for information like date formats)
- people borrow machines from friends, they use them in internet cafes - in these cases the inferred locale may be inappropriate
- always allow the user to select the appropriate language (and locale) from whatever page they are looking at!

68



► How to set up language preferences in your browser

69

- In Internet Explorer:
Tools > Internet Options > General > Languages
- Note: if using a language like fr-CH, add fr too !
- Could add zh-hans and zh-hant



For other user agents, see <http://www.w3.org/International/questions/qa-lang-priorities.html>



Styling using the language attribute

70



► Four possible approaches, in theoretically preferred order

71

- the :lang() pseudo-class selector
- a [lang |= "..."] selector that matches the beginning of the value of a language attribute
- a [lang = "..."] selector that exactly matches the value of a language attribute
- a generic class or id selector



► Using :lang()

72

```
body { font-family: "Times New Roman", serif; }
:lang(ar) { font-family: "Traditional Arabic", serif; font-size: 1.2em; }
:lang(zh-Hant) { font-family: PMingLiU, MingLiU, serif; }
:lang(zh-Hans) { font-family: SimSun-18030, SimHei, serif; }
:lang(din) { font-family: "Doulos SIL", serif; }
```

```
<p>It is polite to welcome people in their own language.</p>
<ul>
<li xml:lang="zh-Hans" lang="zh-Hans">欢迎</li>
<li xml:lang="zh-Hant" lang="zh-Hant">歡迎</li>
<li xml:lang="el" lang="el">Καλοσπασίτε</li>
<li xml:lang="ar" lang="ar">مرحباً</li>
<li xml:lang="ru" lang="ru">Добро пожаловать</li>
<li xml:lang="din" lang="din">Kudua!</li>
</ul>
```

- en-GB in lang attribute matches en in :lang()
- ideal approach, but not yet supported by IE6



▶ Using [lang |= "..."]

```
body (font-family: "Times New Roman", serif;)
[lang]=ar (font-family: "Traditional Arabic", serif; font-size: 1.2em;)
[lang]=zh-Hant (font-family: PMingLiU,MingLiU, serif;)
[lang]=zh-Hans (font-family: SimSun-18030, SimHei, serif;)
[lang]=din (font-family: "Doulos SIL", serif;)
```

```
<p>It is polite to welcome people in their own language:</p>
<ul>
<li xml:lang="zh-Hans" lang="zh-Hans">欢迎</li>
<li xml:lang="zh-Hant" lang="zh-Hant">歡迎</li>
<li xml:lang="el" lang="el">Καλοσωπιατε</li>
<li xml:lang="ar" lang="ar">رحبا</li>
<li xml:lang="ru" lang="ru">Добро пожаловать</li>
<li xml:lang="din" lang="din">Kuduak</li>
</ul>
```

- en-GB in lang attribute matches en in :lang()
- not yet supported by IE6
- case sensitive in Opera



73

▶ Using [lang = "..."]

```
body (font-family: "Times New Roman", serif;)
[lang="ar"] (font-family: "Traditional Arabic", serif; font-size: 1.2em;)
[lang="zh-Hant"] (font-family: PMingLiU,MingLiU, serif;)
[lang="zh-Hans"] (font-family: SimSun-18030, SimHei, serif;)
[lang="din"] (font-family: "Doulos SIL", serif;)
```

```
<p>It is polite to welcome people in their own language:</p>
<ul>
<li xml:lang="zh-Hans" lang="zh-Hans">欢迎</li>
<li xml:lang="zh-Hant" lang="zh-Hant">歡迎</li>
<li xml:lang="el" lang="el">Καλοσωπιατε</li>
<li xml:lang="ar" lang="ar">رحبا</li>
<li xml:lang="ru" lang="ru">Добро пожаловать</li>
<li xml:lang="din" lang="din">Kuduak</li>
</ul>
```

- not yet supported by IE6
- requires exact matching of values



74

▶ Using a generic class or id selector

```
body (font-family: "Times New Roman", serif;)
.ar (font-family: "Traditional Arabic", serif; font-size: 1.2em;)
.zht (font-family: PMingLiU,MingLiU, serif;)
.zhs (font-family: SimSun-18030, SimHei, serif;)
.din (font-family: "Doulos SIL", serif;)
```

```
<p>It is polite to welcome people in their own language:</p>
<ul>
<li class="zhs" xml:lang="zh-Hans" lang="zh-Hans">欢迎</li>
<li class="zht" xml:lang="zh-Hant" lang="zh-Hant">歡迎</li>
<li xml:lang="el" lang="el">Καλοσωπιατε</li>
<li class="ar" xml:lang="ar" lang="ar">رحبا</li>
<li xml:lang="ru" lang="ru">Добро пожаловать</li>
<li class="din" xml:lang="din" lang="din">Kuduak</li>
</ul>
```

- supported by all browsers
- takes up additional time and bandwidth



75

- Character sets & encoding
- Identifying language
 - Declaring the language of text
 - Language attribute values
 - Negotiating language with the server
 - Styling using the language attribute
- IDN & IRIs
- East Asian typography



76

- Character sets & encoding
- Identifying language
- IDN & IRIs
 - Internationalized Domain Names
 - The next step: Internationalized Resource Identifiers (IRIs)
- East Asian typography

Internationalized Domain Names

▶ You too can have a domain name like this !

<http://www.国家经济贸易.gov.cn/>

▶ You too can have a domain name like this !



▶ This is not just a 'nice to have'. Native script domain names are:

- ♦ easier to memorize
 - ♦ easier to interpret
 - ♦ easier to transcribe
 - ♦ easier to create
 - ♦ easier to guess
 - ♦ easier to relate to
- =
- ♦ better for business !
 - ♦ better for finding things !
 - ♦ better for communicating !
 - ♦ better for the Web !

81



▶ What you need:

- ♦ a domain name registrar to fix the characters allowed for their country or top level domain
- ♦ a person or organization to register a domain name in a specially encoded format representing a non-ASCII string
- ♦ a user agent that knows how to convert a non-ASCII domain name to the specially encoded format

↙ Mozilla 1.4 / Netscape 7.1
Opera 7.2

Internationalized Domain Names, K. Momoi
<http://devedge.netscape.com/viewsource/2003/idn/>

82



▶ How it works:

- ♦ user clicks on hyperlink or enters the URI in the address bar of a user agent
`http://www.国家经济贸易.nu/`
- ♦ UA converts string to Unicode and normalizes text
uppercase to lowercase
normalize alternative representations
eg. half-width kana to full
eliminate prohibited characters
eg. space
etc.
- ♦ UA converts 8-bit characters in Unicode to 7-bit Punycode & prepends special marker
`http://www. xn--vcsu3io1ksphtxpyt.nu/`
- ♦ UA sends the request for the page
GET / HTTP/1.1
Host: www.xn--vcsu3io1ksphtxpyt.nu
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.5a) Gecko/20030728 Mozilla Firebird/0.6.1
...

83



▶ Further facts...

- ♦ the domain name is actually still registered using a subset of ASCII characters
- ♦ approved by IETF in March 2003
- ♦ already works on Mozilla 1.4 / Netscape 7.1 and Opera 7.2
- ♦ defined in RFCs 3490, 3491, 3492 and 3454
- ♦ based on Unicode 3.2

84



▶ Further facts...

- all but the standard ACE prefix "xn--" are now disallowed in IDN (eg. RACE "bq--")
- ICANN published an international guideline on the use of IDN characters
- .jp adopted IDN on 10 July 2003

85



The next step:
Internationalized Resource
Identifiers (IRIs)

86



▶ Allowing for non-ASCII pathnames

[http://www.国家经济贸易.gov.cn/
政府信息之窗/政府工作报告.html](http://www.国家经济贸易.gov.cn/政府信息之窗/政府工作报告.html)

87



▶ Issues

- encoding of resource names on servers can be in many different encodings – not just Punycode as for IDN
- current URI standard allows for non-ASCII characters to be encoded as %HH-escaped byte sequences, but doesn't define or carry information about the encoding
- if the file is moved to a location using a different encoding, the non-ASCII parts of URIs may no longer work

88



▶ How it works (simple version):

89

- user clicks on or types in an IRI in any encoding (depends on page)


```
http://www.example.com/März/
März in utf-8 = 4D C3 A4 72 7A
März in Latin1 = 4D E4 72 7A
```
- UA converts to UTF-8 and converts non-ASCII characters to %HH escapes


```
http://www.example.com/M%C3%A4rz/
```
- protocol passes information through in its normal form (but must receive UTF-8)


```
GET /M%E4rz/ HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.5a) Gecko/20030728 Mozilla Firebird/0.6.1
...
```
- request hits the server
 - if filestore is in UTF-8, server simply accesses the resource
 - if filestore in another encoding, server needs to convert from UTF-8
 - see, for example, Martin Dürst's fileiri apache module
 - <http://www.w3.org/International/resources.html#tools>



▶ Further facts...

90

- IRI spec is an IETF Internet-Draft
- Internet Explorer options include "Always send URLs as UTF-8"
- already works on latest Mozilla / Netscape and Opera browsers
- an increasing number of protocols accept UTF-8
- various documents already support IRIs – eg. XML 1.0 system identifiers, XLink href attribute, XMLSchema anyURI datatype, HTML 4.0

IRI Draft specification
<http://www.w3.org/International/iri-edit/>



- Character sets & encoding
- Identifying language
- IDN & IRIs
 - Internationalized Domain Names
 - The next step: Internationalized Resource Identifiers (IRIs)
- East Asian typography

91



- Character sets & encoding
- Identifying language
- IDN & IRIs
- East Asian typography
 - Lists
 - CSS3 Text module
 - CSS3 Fonts module
 - Ruby
 - Making it happen

92



Lists

93



Lists

▶ CSS2 and CSS2.1

- **list-style-type (CSS2)**
disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-alpha | lower-latin | upper-alpha | upper-latin | hebrew | armenian | georgian | cjk-ideographic | hiragana | katakana | hiragana-iroha | katakana-iroha | none | inherit
- **list-style-type (CSS2.1)**
disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-latin | upper-latin | none | inherit

Push for adoption, so you don't lose these features!

- 一. 供HTML和XML使用的字符输入模型
 - 二. 字符输入的分辨和脱离
 - 三. 语言转换
 - 四. 国际化域名
 - 五. 国际化的资源标识符
- あ. 出願資格認定書交付申請書(所定用紙)
い. 当該学校の教育が12年の課程であることを証明できるもの
う. 当該学校の成績証明書(または調査書)又は卒業証明書または卒業見込み証明書
お. 当該学校の教育内容を証明できるもの...
- い. 出願資格認定書交付申請書(所定用紙)
ろ. 当該学校の教育が12年の課程であることを証明できるもの
は. 当該学校の成績証明書(または調査書)又は卒業証明書または卒業見込み証明書
ほ. 当該学校の教育内容を証明できるもの...
- ア. 出願資格認定書交付申請書(所定用紙)
イ. 当該学校の教育が12年の課程であることを証明できるもの
ウ. 当該学校の成績証明書(または調査書)又は卒業証明書または卒業見込み証明書
エ. 卒業証明書または卒業見込み証明書
オ. 当該学校の教育内容を証明できるもの...

94



Lists

▶ CSS3 Lists Module

- **list-style-type (algorithmic)**
armenian | cjk-ideographic | ethiopic-numeric | georgian | hebrew | japanese-formal | japanese-informal | lower-armenian | lower-roman | simp-chinese-formal | simp-chinese-informal | syriac | tamil | trad-chinese-formal | trad-chinese-informal | upper-armenian | upper-roman
- **list-style-type (numeric)**
arabic-indic | binary | bengali | cambodian | decimal | decimal-leading-zero | devanagari | gujarati | gurmukhi | kannada | khmer | lao | lower-hexadecimal | malayalam | mongolian | myanmar | octal | oriya | persian | telugu | tibetan | thai | upper-hexadecimal | urdu
- **list-style-type (alphabetic)**
afar | amharic | amhare-abegede | cjk-earthly-branch | cjk-heavenly-stem | ethiopic | ethiopic-abegede | ethiopic-abegede-am-et | ethiopic-abegede-gez | ethiopic-abegede-ti-er | ethiopic-abegede-ti-et | ethiopic-halehame-aa-er | ethiopic-halehame-aa-et | ethiopic-halehame-am-et | ethiopic-halehame-gez | ethiopic-halehame-om-et | ethiopic-halehame-sid-et | ethiopic-halehame-so-et | ethiopic-halehame-ti-er | ethiopic-halehame-ti-et | ethiopic-halehame-tig | hangul | hangul-consonant | hiragana | hiragana-iroha | katakana | katakana-iroha | lower-alpha | lower-greek | lower-norwegian | lower-latin | oromo | sidama | somali | tigre | tigrinya-er | tigrinya-er-abegede | tigrinya-et | tigrinya-et-abegede | upper-alpha | upper-greek | upper-norwegian | upper-latin
- **list-style-type (non-repeating)**
circled-decimal | circled-lower-latin | circled-upper-latin | dotted-decimal | double-circled-decimal | filled-circled-decimal | parenthesised-decimal | parenthesised-lower-latin

95



CSS3 Text Module

96



Text layout

- `direction`
lr | rl
- `block-progression`
tb | rl | lr
- `writing-mode` [shortcut]
lr-tb | rl-tb | tb-lr
- `glyph-orientation-vertical`
<angle> | auto | upright | inline
- etc...

世界的に話すなら、Unicode
です。第10回のUnicode会議
は一九九七年三月十日〜十二
日、ドイツのマインツで開か
れます。参加希望の方は今す
ぐ登録してください。この会
議では、グローブ

97

W3C®

Text alignment & justification

- `text-justify`
auto | inter-word | inter-ideograph |
distribute | newspaper | inter-cluster |
kashida
- `text-justify-trim`
none | punctuation | punctuation-
and-kana
- etc...

請用統一碼 (Uni
code) 你現在

展(北

展(北

98

W3C®

Line breaking

- `line-break`
normal | strict
- `word-break-cjk`
normal | break-all | keep-all
- `word-break-inside`
normal | hyphenate
- `word-break`
<'word-break-cjk'> || <'word-break-inside'>
- etc...

ではソフトウ
エアの国際化

ではソフト
ウェアの国際

99

W3C®

Line breaking

- `line-break`
normal | strict
- `word-break-cjk`
normal | break-all | keep-all
- `word-break-inside`
normal | hyphenate
- `word-break`
<'word-break-cjk'> || <'word-break-inside'>
- etc...

日在德国
Mainz 市举行

日在德国 Mai
nz市举行的第

100

W3C®

▶ Text spacing

- punctuation-trim
none | start
- text-autospace
none | [ideographic-numeric || ideograph-alpha || ideograph-parenthesis || ideograph-space]
- etc...

経験分 (万维)

第10回のUnicode会議

▶ Document grid

- line-grid-mode
none | ideograph | all
- line-grid-progression
text-height | line-height | <length>
- line-grid
<'line-grid-mode' || 'line-grid-progression'>
- etc...

参加希望の方は今すぐ登録して会議では、グローバルなインターネット、E-mail、ソフトウェアの国際化およびローカリゼーション、OSおよびアプリケーションでのUnicodeのインプリメンテーション、フロント、テキスト表示、マルチ言語コンテンツエンジンにおける業界の専門家が集まります。

▶ Document grid

- line-grid-mode
none | ideograph | all
- line-grid-progression
text-height | line-height | <length>
- line-grid
<'line-grid-mode' || 'line-grid-progression'>
- etc...

参加希望の方は今すぐ登録して会議では、グローバルなインターネット、Unicode、ソフトウェアの国際化およびローカリゼーション、OSおよびアプリケーションでのUnicodeのインプリメンテーション、フロント、テキスト表示、マルチ言語コンテンツエンジンにおける業界の専門家が集まります。

▶ Miscellaneous text formatting

- hanging-punctuation
none | start | end | both
- text-combine
none | letters | lines
- etc...

可使用的网页。这个导课
可使用的网页。这个导课会讲

割注 (これはわりちゅうです) です。

くみもじくみもじ

CSS3 Fonts Module

106



CSS3 Fonts Module

▶ Font emphasis

107

- `font-emphasize-style`
none | accent | dot | circle | disc
- `font-emphasize-position`
before | after
- `font-emphasize [shorthand]`
<font-emphasize-style> ||
<font-emphasize-position>
- etc...

これは日本語の文章です。
これは日本語の文章です。



Ruby

108



Ruby

▶ Who is Ruby?

109

振り仮名



▶ Who is Ruby?

110

ふ が な
振り仮名
ルビ

W3C®

▶ Ruby markup

111

```
<ruby>
  <rb></rb>
  <rt></rt>
</ruby>
```

W3C®

▶ Ruby markup

112

```
<ruby>
  <rb>紙芝居</rb>
  <rt></rt>
</ruby>
```

W3C®

▶ Ruby markup

113

```
<ruby>
  <rb>紙芝居</rb>
  <rt>かみしばい</rt>
</ruby>
```

<p>これは<ruby><rb>紙芝居</rb><rt>かみしばい</rt></ruby>です。</p>

W3C®

- ▶ Any character is valid as ruby text.

114

中国话

zhong guo hua

N E C
日本電気

計 算 機
コンピュータ

ガバメント・セクション
民 政 局

W3C®

- ▶ Ruby styling

115

- `ruby-position`
before | after | right
- `ruby-align`
auto | start | left | center | end | right
- `ruby-overhang`
auto | start | end | none
- `ruby-span`
attr(x) | none

かみしばい
紙芝居

かみしばい
紙芝居

W3C®

- ▶ Ruby styling

116

- `ruby-position`
before | after | right
- `ruby-align`
auto | start | left | center | end | right
- `ruby-overhang`
auto | start | end | none
- `ruby-span`
attr(x) | none

紙芝居
かみしばい

紙芝居
かみしばい

W3C®

- ▶ Ruby styling

117

- `ruby-position`
before | after | right
- `ruby-align`
auto | start | left | center | end | right
- `ruby-overhang`
auto | start | end | none
- `ruby-span`
attr(x) | none

第十、届
第十、届

W3C®

▶ Ruby styling

119

- `ruby-position`
before | after | right
- `ruby-align`
auto | start | left | center | end | right
- `ruby-overhang`
auto | start | end | none
- `ruby-span`
attr(x) | none

うきよえ むかしばなし
浮世絵 昔話

W3C®

▶ Ruby styling

120

- `ruby-position`
before | after | right
- `ruby-align`
auto | start | left | center | end | right
- `ruby-overhang`
auto | start | end | none
- `ruby-span`
attr(x) | none

むかしばなし
いい昔話だ

W3C®

Making it happen

121

W3C®

▶ Some of these features already work to some degree on some browsers

122

- however, still browser specific (many things work on Internet Explorer)
- typically implementations based on very early versions of the standards – check carefully against the standards
- still, it gives a good idea of how the Web could be
- link to demo page
<http://people.w3.org/rishida/scripts/samples/>

W3C®

▶ **Ways you can get involved and help support the needs of your local language on the Web**

123

- ◆ participate in the development of the specifications to input Chinese (or other) requirements
- ◆ lobby developers of user agents and editing tools to incorporate the new features being standardized
- ◆ help review specifications for internationalization issues
- ◆ implement new features in user agents during Candidate Recommendation phase, and beyond
- ◆ use the new features as they become available



- Character sets & encoding
- Identifying language
- IDN & IRIs
- **East Asian typography**
 - Lists
 - CSS3 Text module
 - CSS3 Fonts module
 - Ruby
 - Making it happen

124



- Character sets & encoding
- Identifying language
- IDN & IRIs
- East Asian typography

125



- ▶ Discuss various aspects of content authoring that support universal access to the Web
- ▶ You will understand
 - ▶ how to declare and use characters & character encodings in X/HTML
 - ▶ how to declare the language of the document or parts of the document
 - ▶ how to use Chinese or other scripts in URIs according to the latest standards
- ▶ You will see some of the new style capabilities that will soon be available for Asian languages

126





Thankyou

<http://www.w3.org/International/>