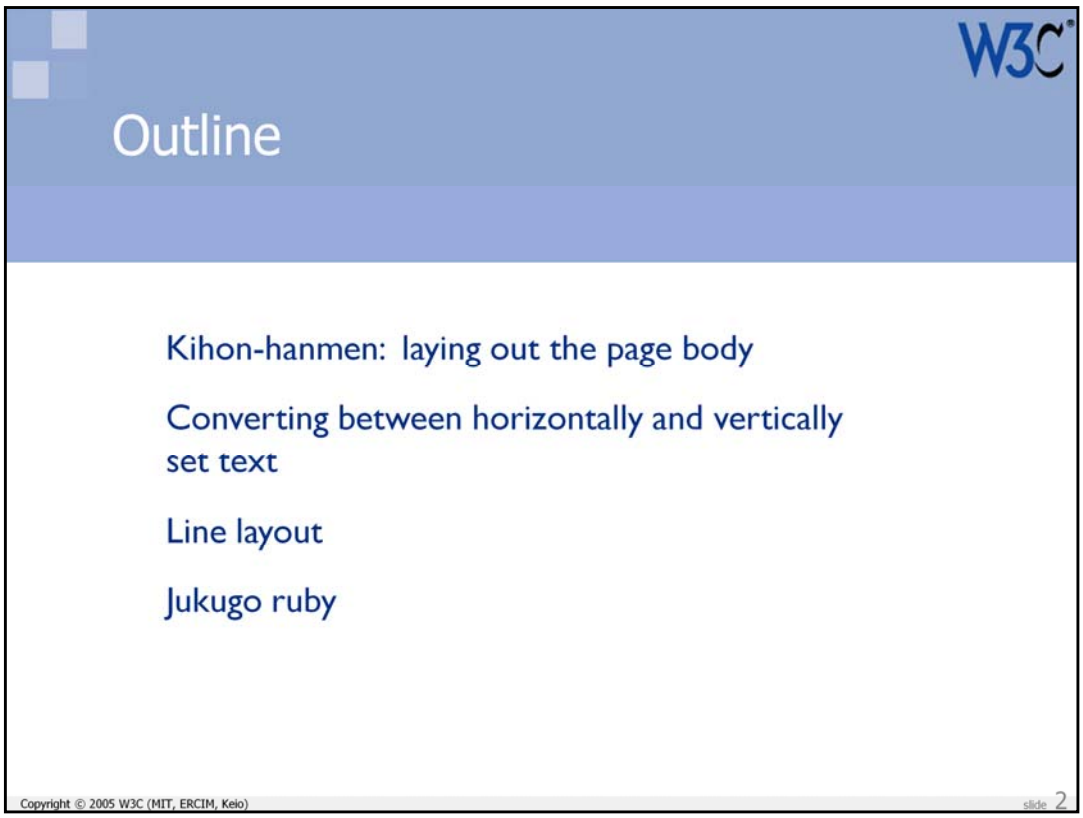
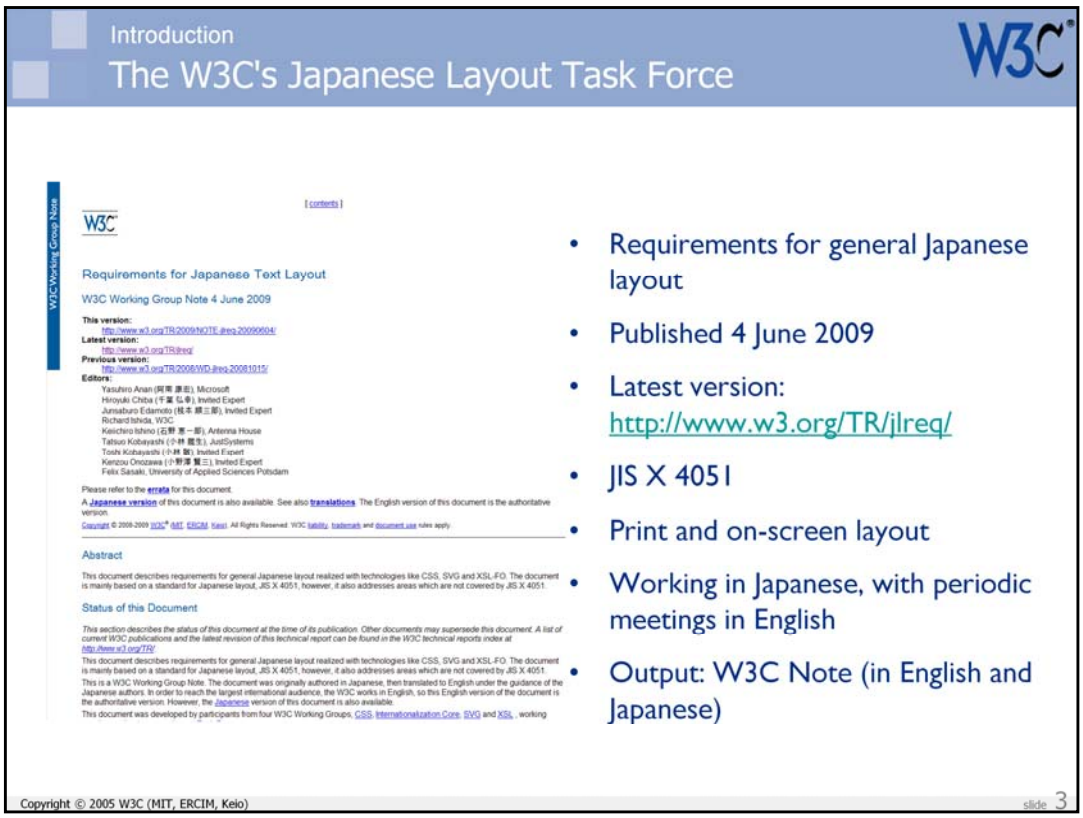
W
3
C
New Work on Japanese Layout Requirements
Richard Ishida, W3C
Based on a presentation by Richard Ishida, Tatsuo Kobayashi, JustSystem and Steve Zilles, Adobe Systems



This presentation describes a document published in June 2009 that describes requirements for Japanese layout.



The goal of the W3C's Japanese Layout Task Force was the creation of a document about requirements for general Japanese layout realized with technologies like CSS, SVG and XSL-FO. The document is mainly based on a standard for Japanese layout, JIS X 4051. However, it addresses also areas which are not covered by JIS X 4051.

The requirements are not expressed on a technology-specific level. It is assumed that the participating working groups (see below) will decide independently how (and if) to implement the requirements in their specifications.

The document was developed in Japanese by Japanese participants, but with periodic input from English speakers. We published in both Japanese and English. Because the W3C is an international organization that works in the lingua franca of English, the English version is authoritative. This dual language development was a first for the W3C.

Introduction

The W3C's Japanese Layout Task Force

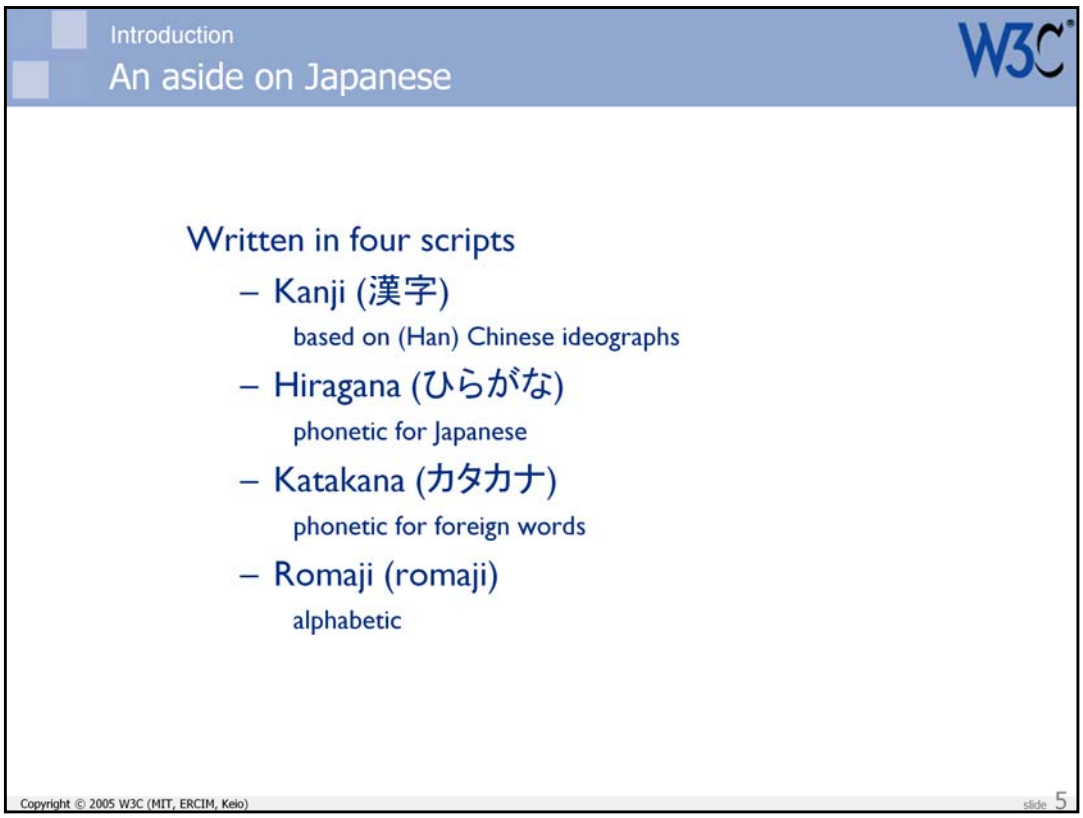
Editors/authors:

Yasuhiro Anan (阿南 康宏), Microsoft
Hiroyuki Chiba (千葉 弘幸), Invited Expert
Junsaburo Edamoto (枝本 順三郎), Invited Expert
Richard Ishida, W3C
Keiichiro Ishino (石野 恵一郎), Antenna House
Tatsuo Kobayashi (小林 龍生), JustSystems
Toshi Kobayashi (小林 敏), Invited Expert
Kenzou Onozawa (小野澤 賢三), Invited Expert
Felix Sasaki, University of Applied Sciences Potsdam

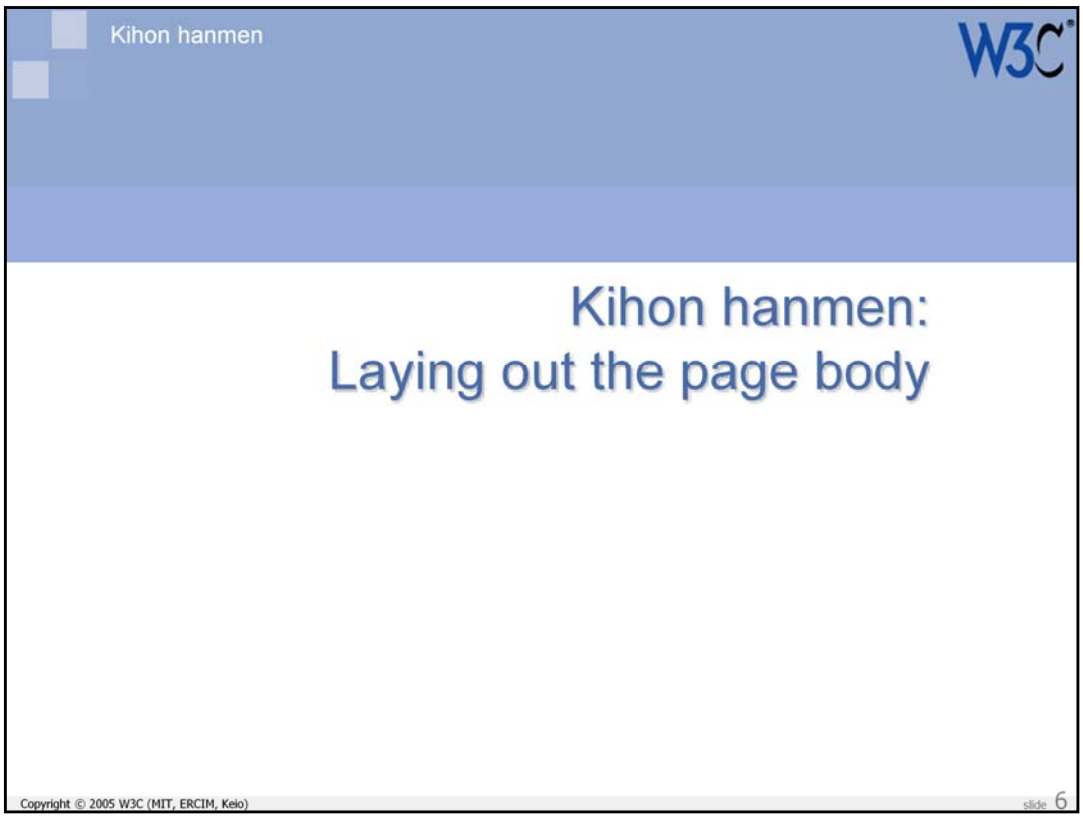
- Also participants from Internationalization, CSS, SVG, XSL Working Groups

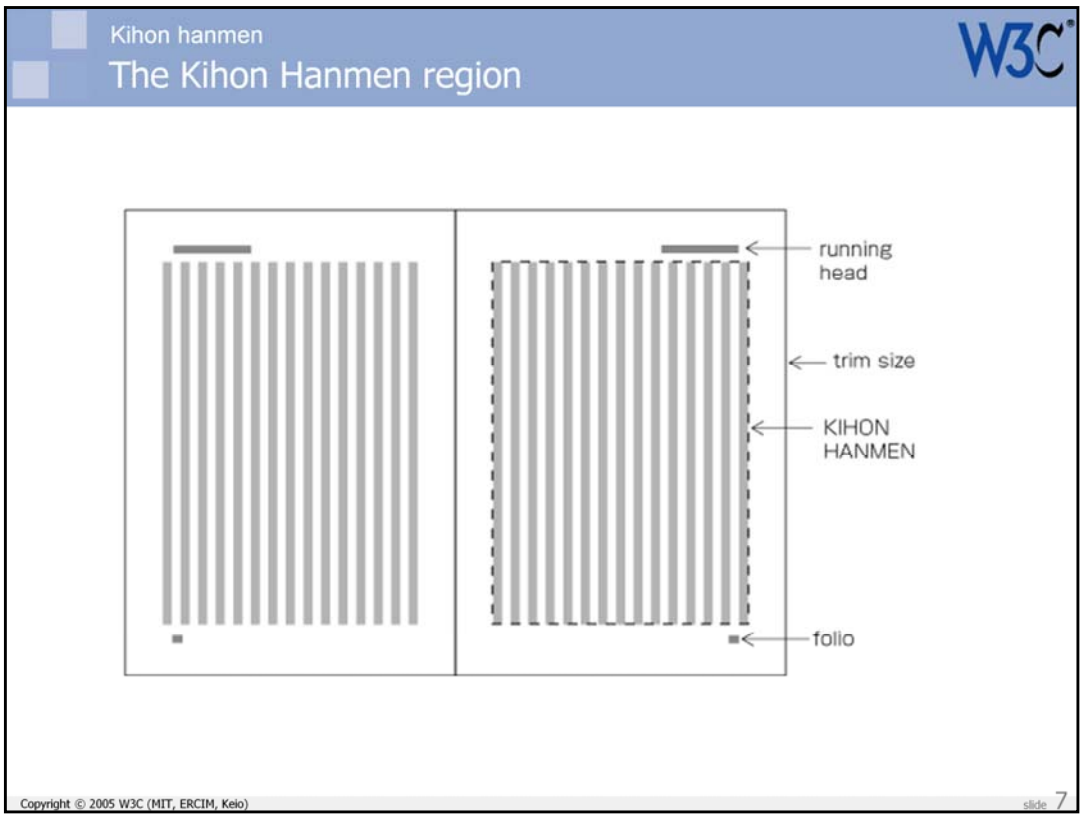
Copyright © 2005 W3C (MIT, ERCIM, Keio) slide 4

In addition to the Japanese core, the work drew on participation from the W3C Internationalization, CSS, SVG and XSL Working Groups.



Japanese is a multisyllabic language that is very different that mono syllabic, tonal Chinese, but lacking a script of their own, the Japanese initially adopted the Chinese ideographic script to write Japanese. When this proved not adequate to the task they added two phonetic scripts (first hiragana and then katakana). Today, hiragana is used to express things like verb and adjective inflections (endings) (lacking in Chinese) and to write words that lacked Chinese character(s). And, katakana is used to express words imported in the 19th century or later and for proper names and/or emphasis. Finally, the Latin alphabet is used to write Japanese in a form that foreigners might understand.

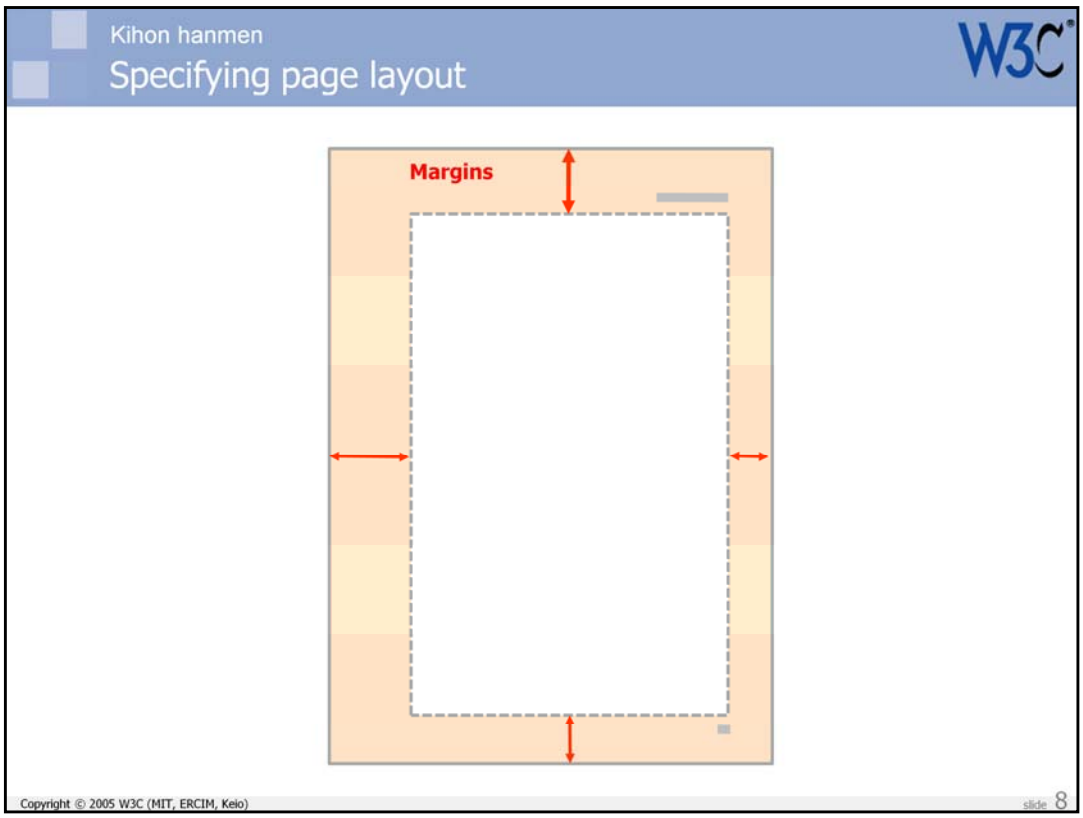

Kihon hanmen
Kihon hanmen:
Laying out the page body
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 6



The kihon-hanmen is a template for the area of the page that contains the main body of the Japanese text. It doesn't include page headers and footers.

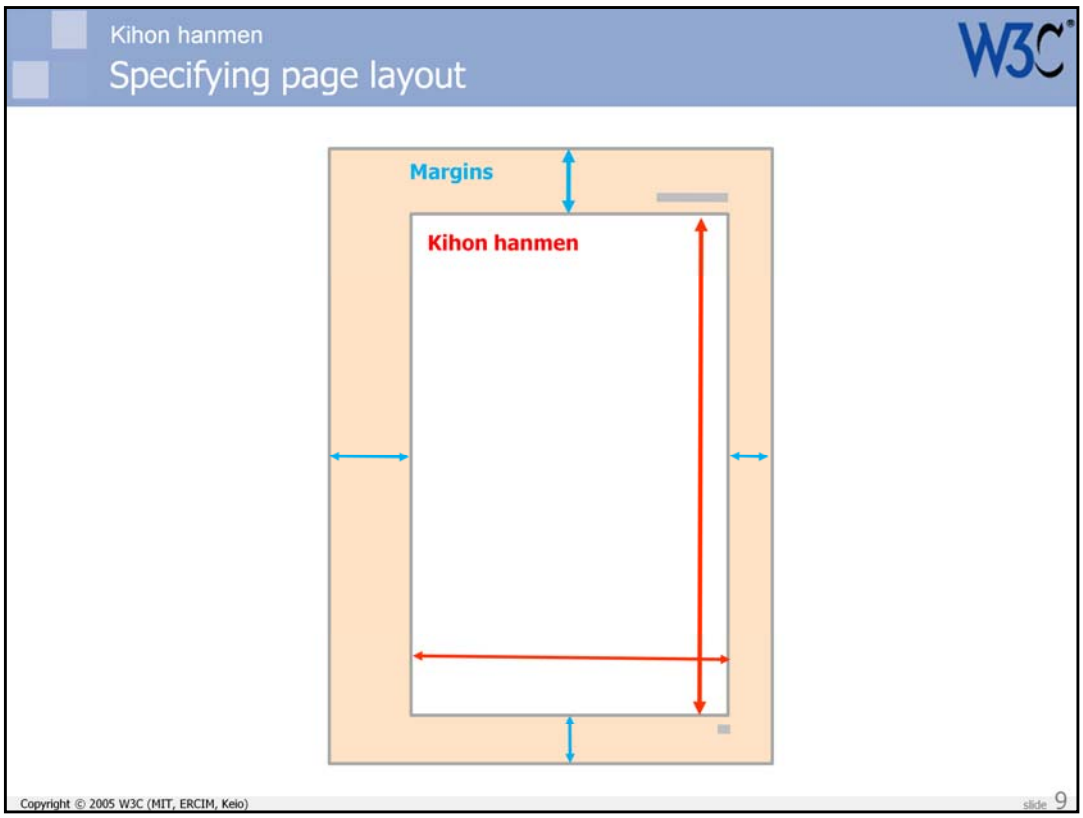
The document in the slide is vertically set and typical of the layout of a Japanese novel.

In order to understand why the kihon-hanmen is of interest, we must first understand some of the characteristics of Japanese characters and their use.



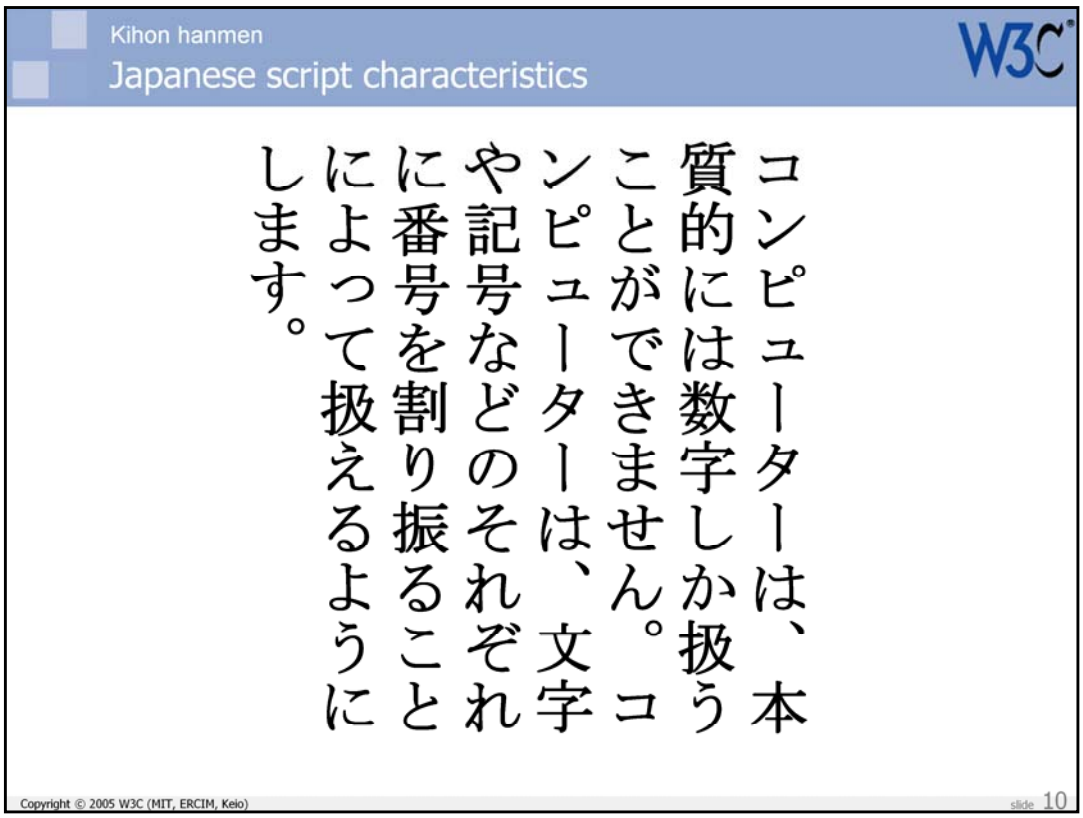
A typically Western way of creating the main text region on a page involves just specification of margins. For example, to create a simple document in XSL-FO, after specifying the printable area of the page using page margins, you specify the size, within the printable area, of the main body region by specifying its margins. (Those margins can then be used for page headers and footers.)

The key point is that you don't usually specify the size of the body region itself. The size of the body region is whatever is left over after you have specified its margins. In other words, you specify the size of the blank space, then just flow the text into the remaining space.



Because of the regularity in size of Japanese characters, which fit into the same sized square shape, Japanese people think of page layout slightly differently.

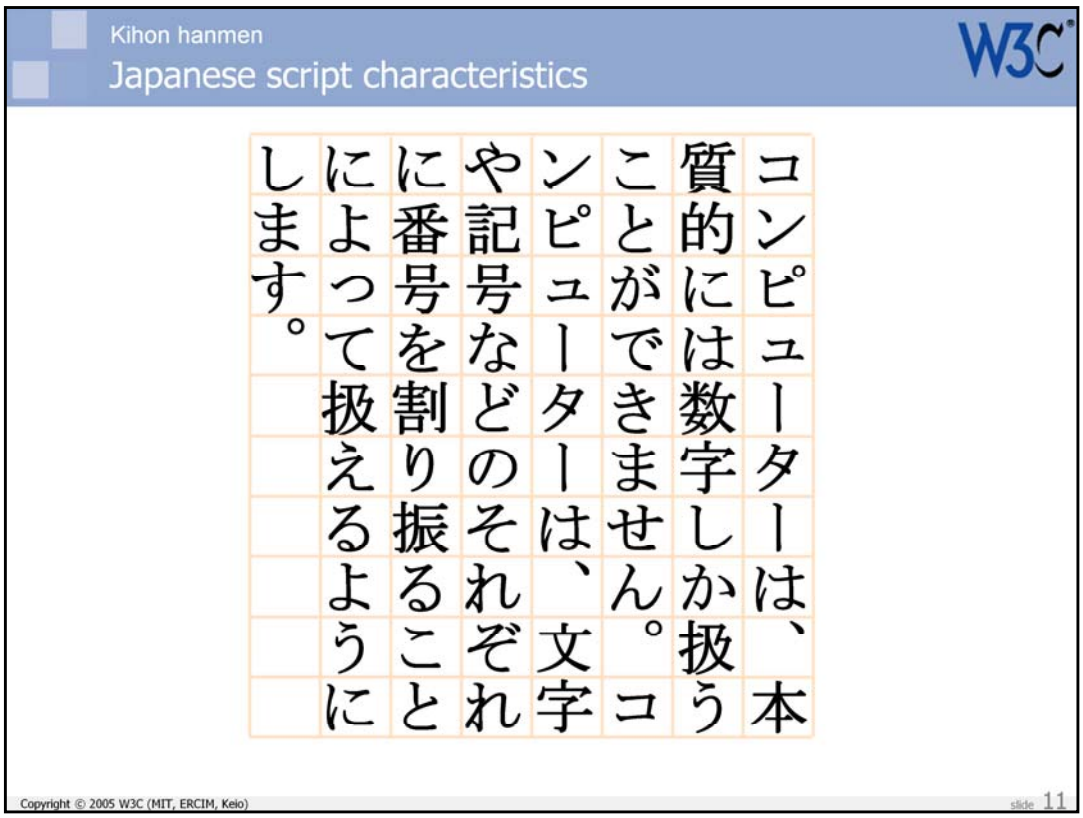
Their natural inclination is to first define the size of the main text area (ie. the kihon hanmen). Only after that are the margins considered.



Individual Japanese kanji, hiragana and katakana characters normally sit in square boxes that are all the same size. This leads to a very regular effect.

Even characters such as the Japanese period and comma shown on the slide typically sit within a square box of the same size.

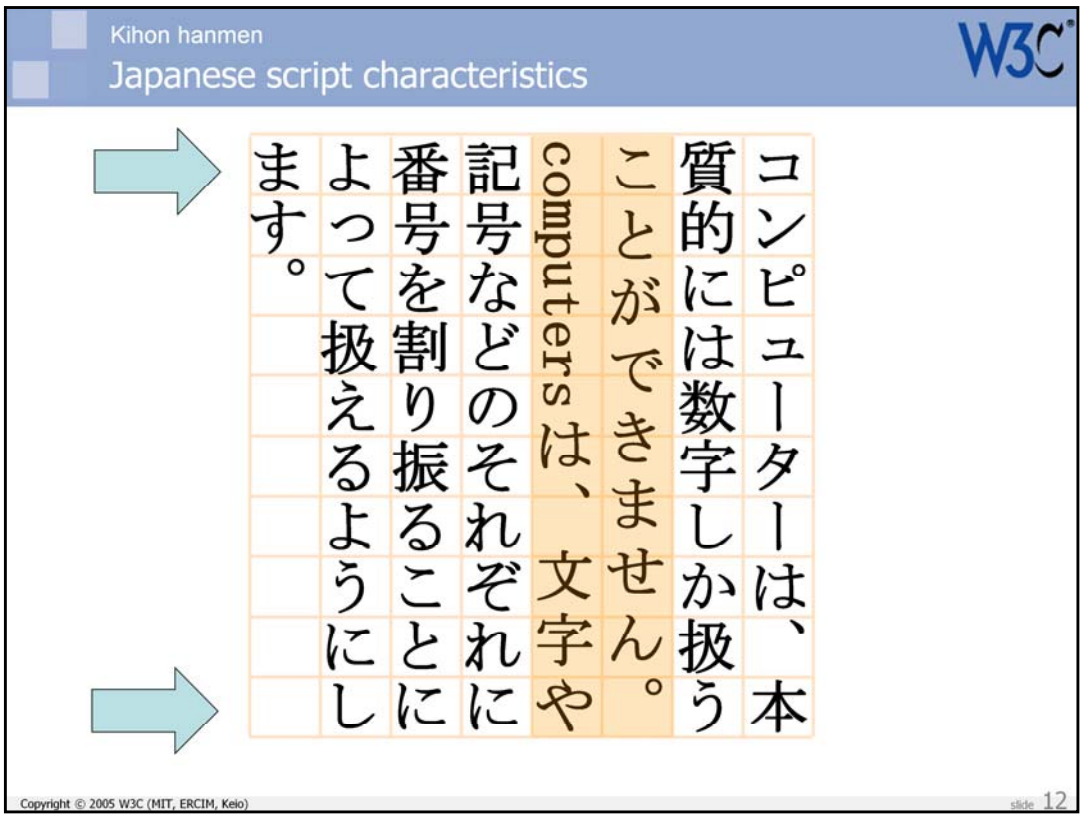
There are no space characters between characters in the slide above.



The regularity of the character sizes can produce a grid-like effect. Lines could be drawn through the text both vertically and horizontally, as shown.

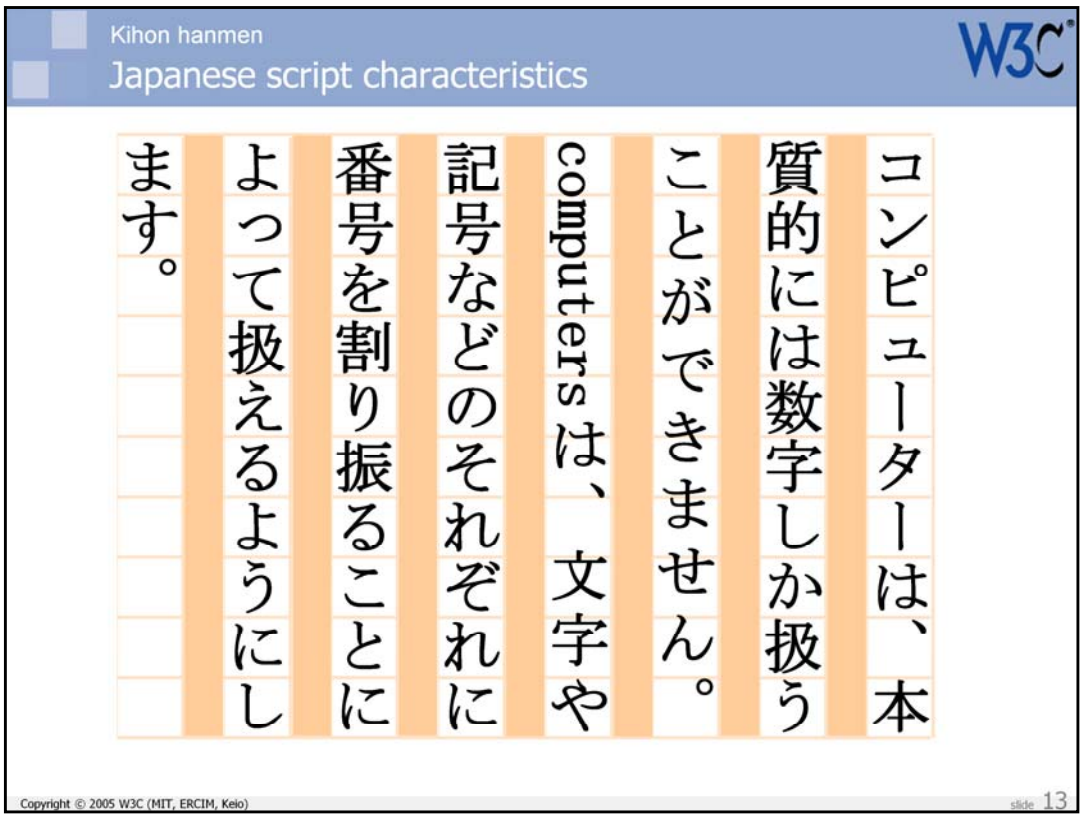
This is an important feature of the kihon-hanmen. Rather than being simply a rectangular area on the page that will be filled with text, the dimensions of the kihon-hanmen are specified by indicating the height and width in terms of a number of characters. This establishes a kind of logical grid which will be used to position characters within the kihon-hanmen.

What is shown on this slide is a very simplistic view of such a grid, with cells the same size as the square characters that fit within the kihon-hanmen. In reality, things are a little more complicated, as we'll see on the following slides.

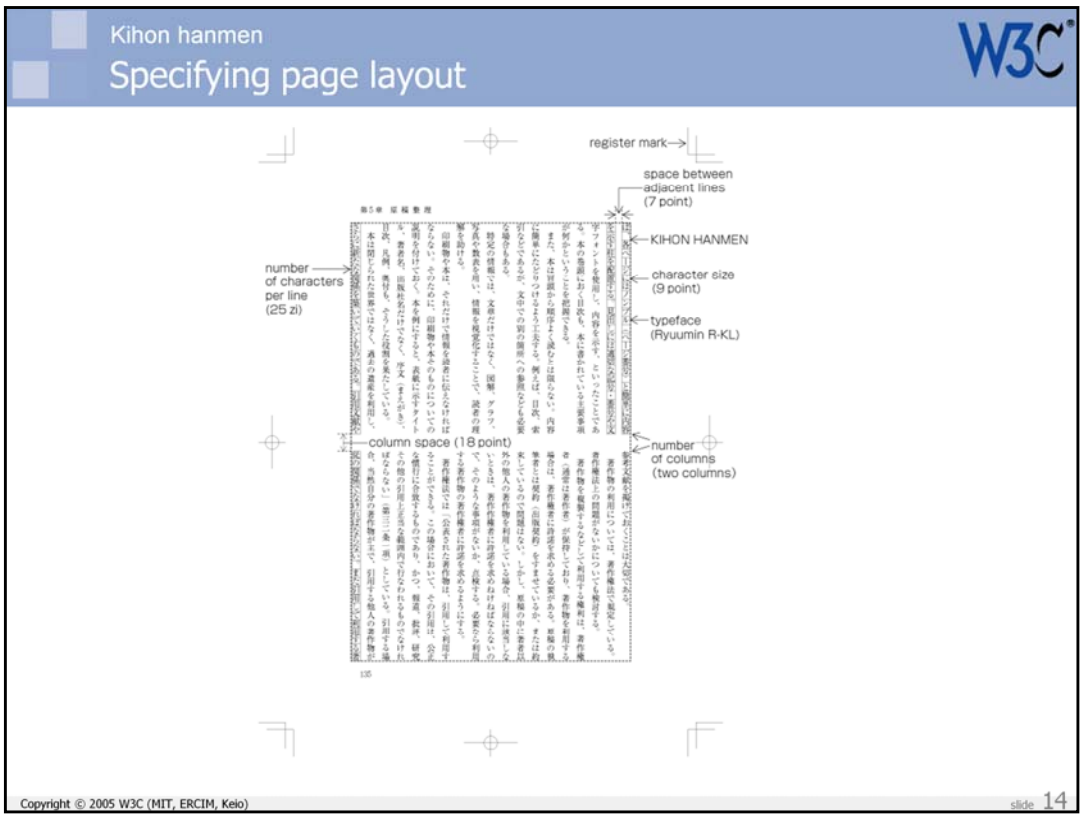


Embedded Latin characters, rules about what characters can begin or end a line, etc, tend to produce small variations in the placement of characters relative to the grid. This means that characters within a line in the kihon-hanmen don't necessarily fit exactly to the grid layout. Nevertheless, where characters come adrift from the grid cells, there is a strong requirement to end each line with the last character flush with the bottom of the grid. This is achieved using rules for adjusting space on the line.

For this reason, it may be more accurate to envisage the kihon-hanmen as a series of slots rather than a grid.

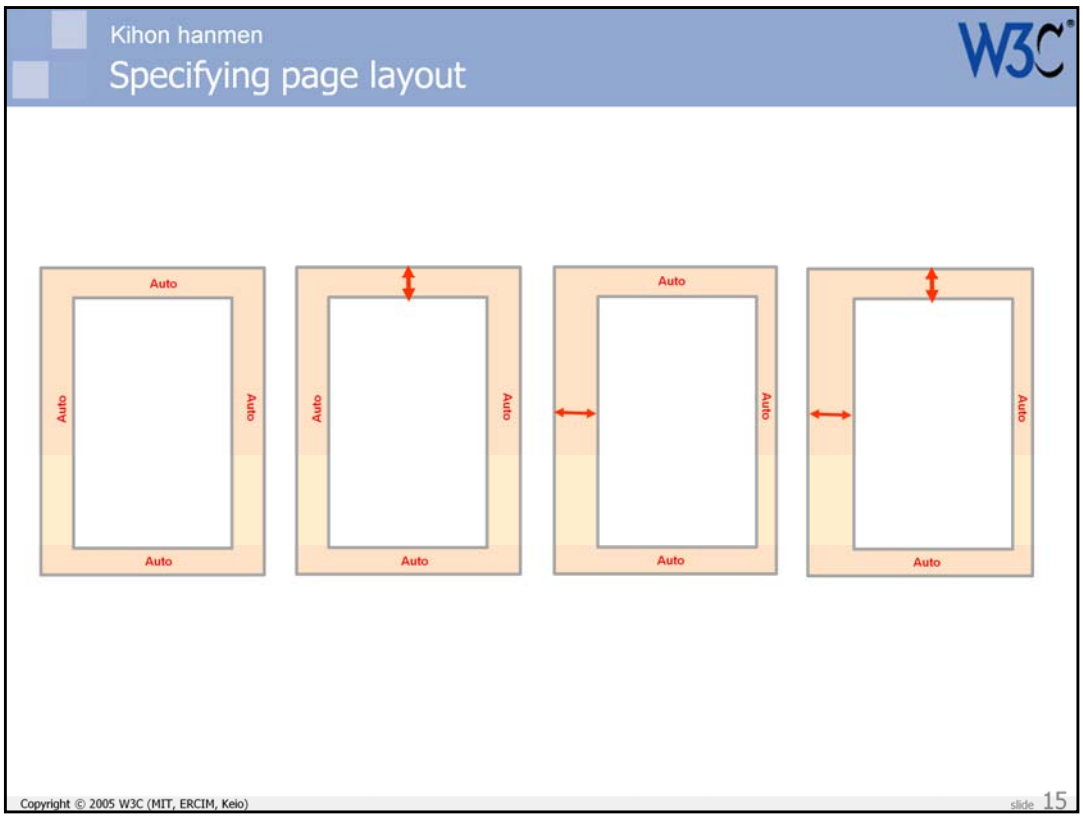


In addition, Japanese text usually has some interline spacing, typically ranging from half a character box in width, for short lines, to a full character box for longer lines.



For a page with a single column, defining the size of the kihon hanmen involves choosing a character size, specifying the length of the lines as a number of characters, then specifying the other dimension of the text area as the number of lines per page and the size of the interline gap. (There are preferred sizes and ratios for characters, line lengths and lines per page.)

For documents with multiple columns, the size of the gap between the columns is also taken into account.



Once the size of the kihon-hanmen has been decided, there are a number of strategies for determining margins.

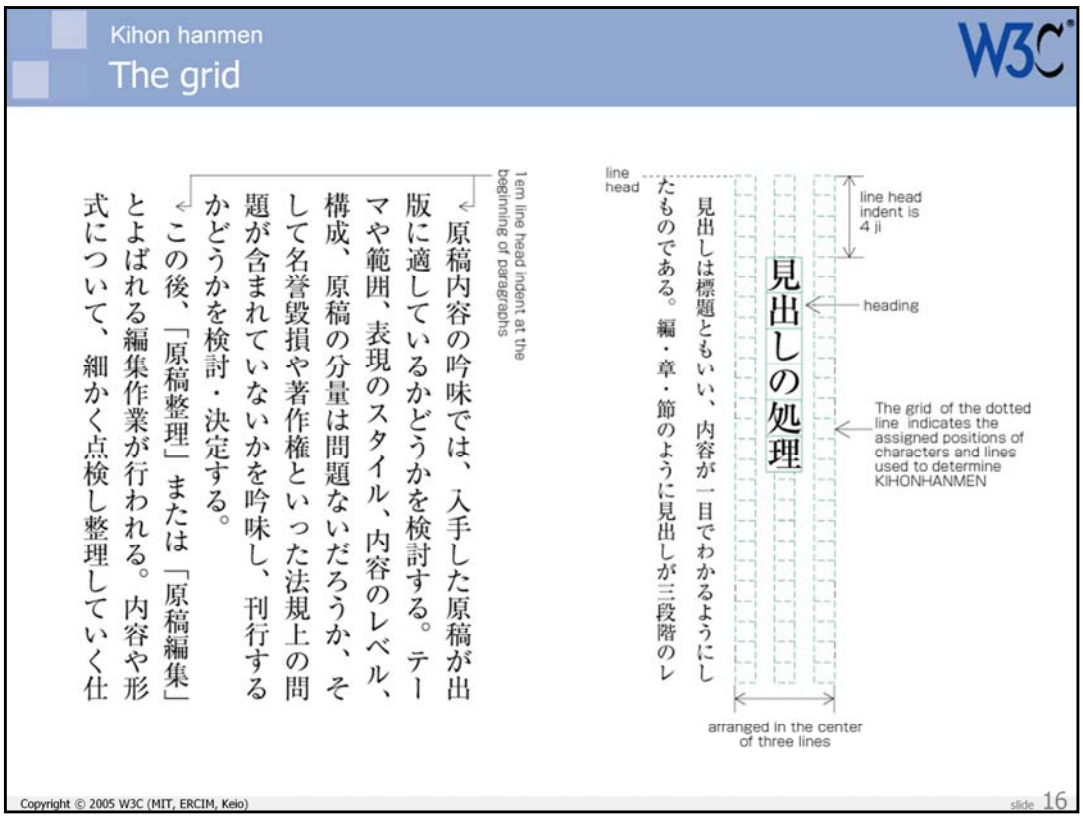
The common default is to center the kihon-hanmen in the body region.

Other alternatives include:

1. Specifying the top space (for horizontally set pages) or the bottom (for vertically set pages), and centering the sides.
2. Specifying the size of the gutter to the side, and centering the top and bottom.
3. A mixture of 1 and 2.

In summary, to specify the position of the main text flow in a simple document using kihon-hanmen you would specify the dimensions of the main text region itself, and zero to two margins.

There has been discussion at the W3C about making it easier to specify the body region in this way in XSL-FO 2.0.

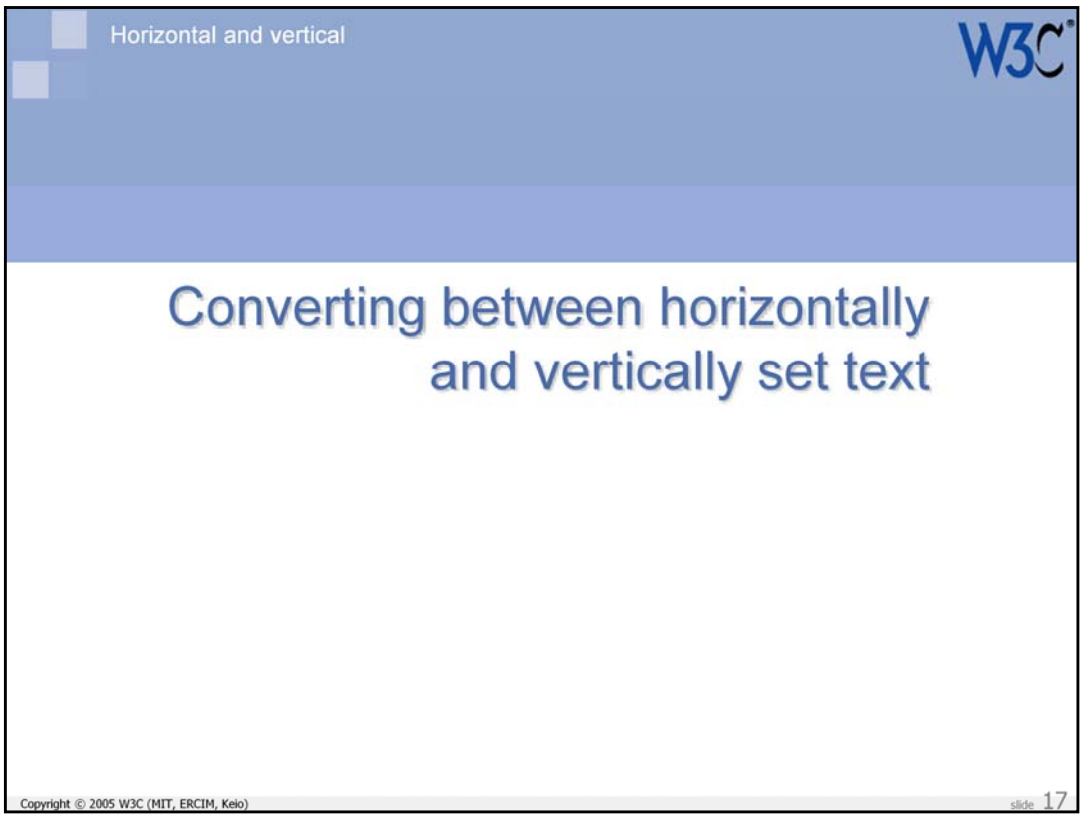


In a number of ways, elements positioned within the kihon-hanmen tend to snap to the logical grid that underlies this region.

For example, the first line of a paragraphs tends to be indented, and the usual indentation is one logical character box.

Headings in the document also tend to be positioned relative to the grid, even though the size of the characters in the title is larger than that used for the logical grid of the kihon-hanmen. In the example shown, the heading is indented by a given number of character sizes, and balanced across three logical lines.

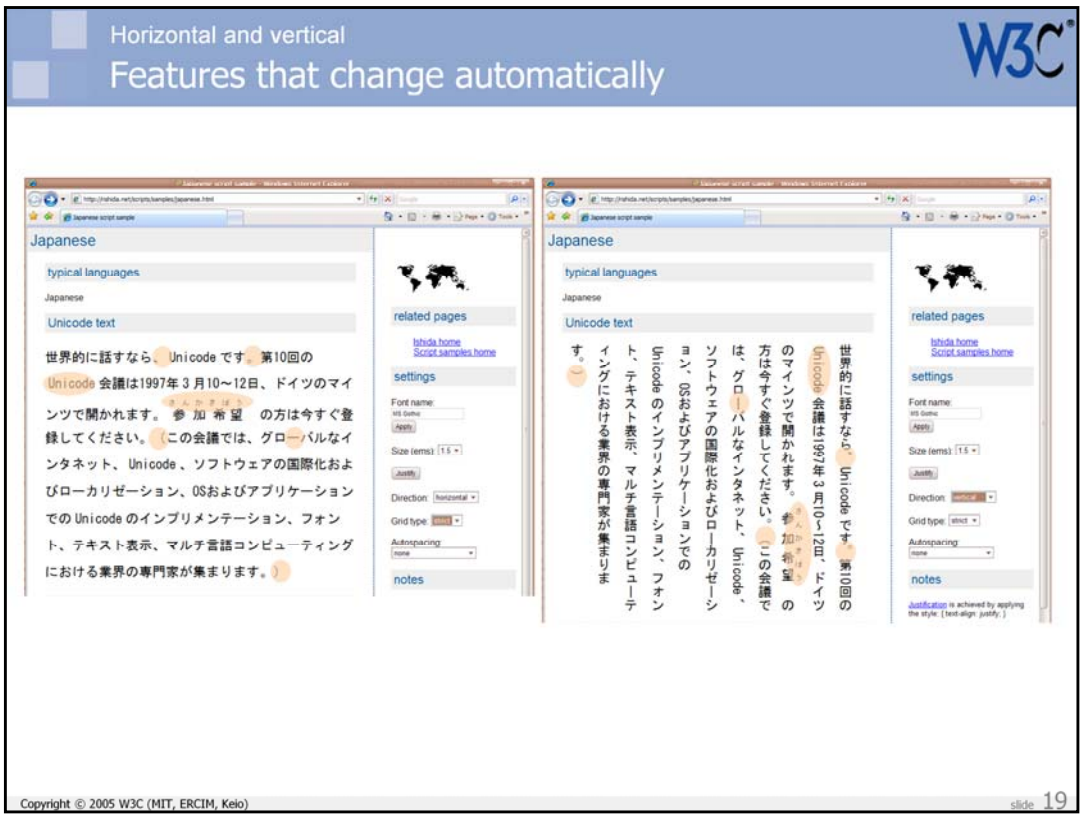
Figures and surrounding text tend to be positioned relative to the kihon-hanmen grid.


Horizontal and vertical
Converting between horizontally and vertically set text
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 17

You may think that you would be able to switch between horizontal and vertical text at the flick of a CSS switch. I even have a page that allows you to do that, using some JavaScript that changes the writing-mode property on a block of Japanese text (see http://rishida.net/scripts/samples/japanese.html - requires Internet Explorer 5.5+).

There are some differences between text in vertical and horizontal layouts that can be dealt with automatically, but there are others that are not so easy to handle.

The change shown on the slide is brought about by simply changing the value of the writing-mode property on the enclosing block from lr-tb (left-to-right block flow, top-to-bottom line progression) to tb-rl (top-to-bottom block flow, right-to-left line progression).



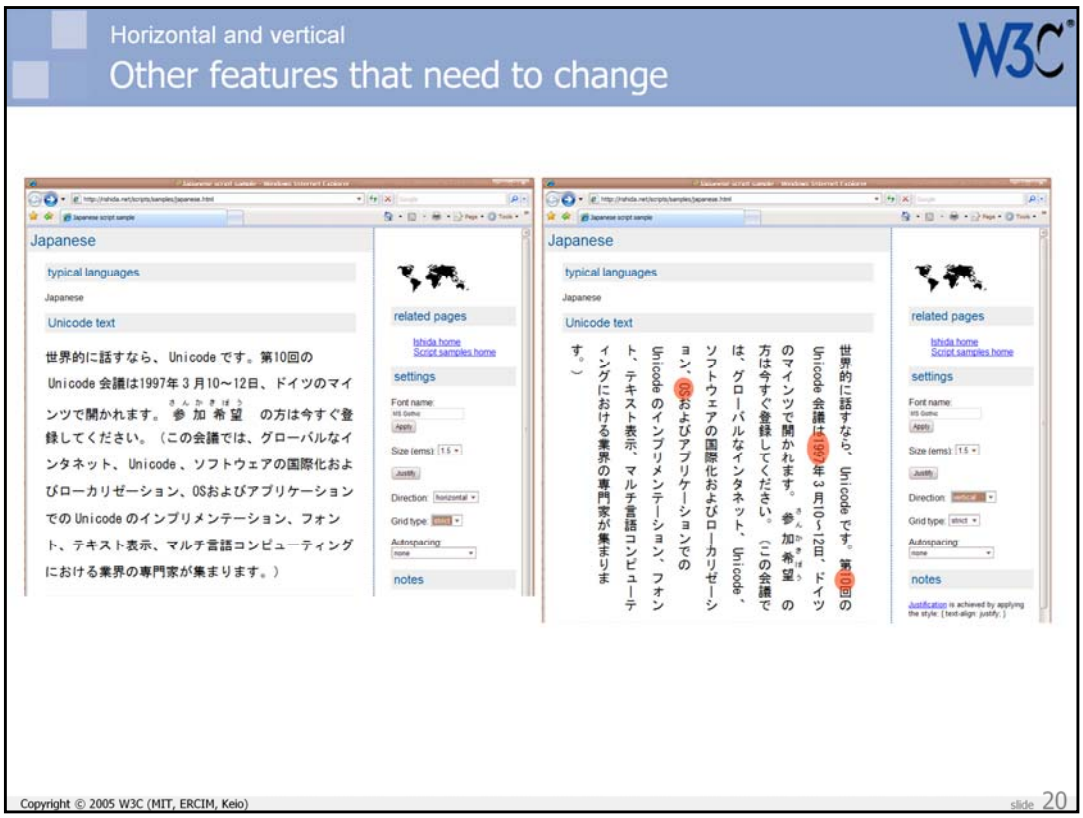
This slide shows some things that already work in Internet Explorer, using this simple switch.

The Japanese comma and full-stop move from the bottom left of the character square to the top right. This is achieved by choosing a vertical-text-specific glyph from the font. The character codes remain the same.

Sound elongation marks and punctuation such as parentheses are rotated.

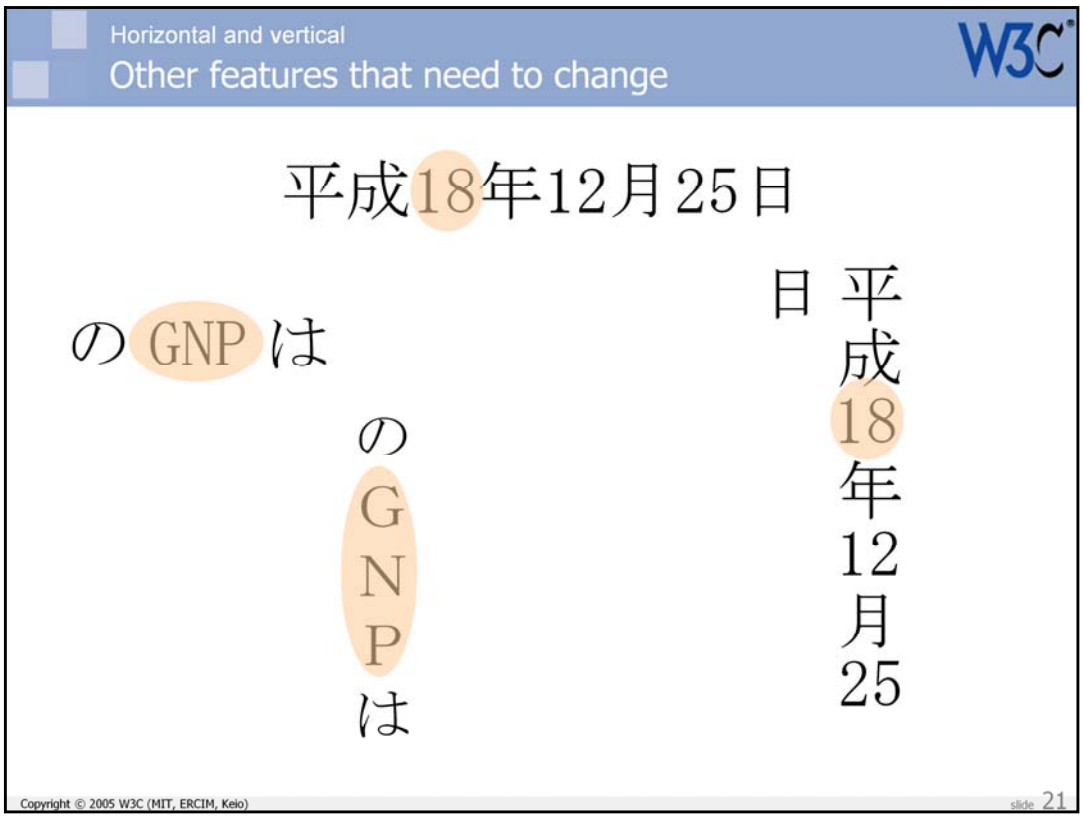
Runs of Latin text flow down the page in vertical lines.

Ruby text moves from above the horizontal line to the right of the vertical line.



Unfortunately, there are other aspects of the text that should have changed, but didn't. For example, some numbers would have looked better as kanji characters, rather than Western digits, in vertical text; other numbers should run horizontally within the vertical line progression; and some Latin text that is used as an acronym would look better if the characters were not rotated.

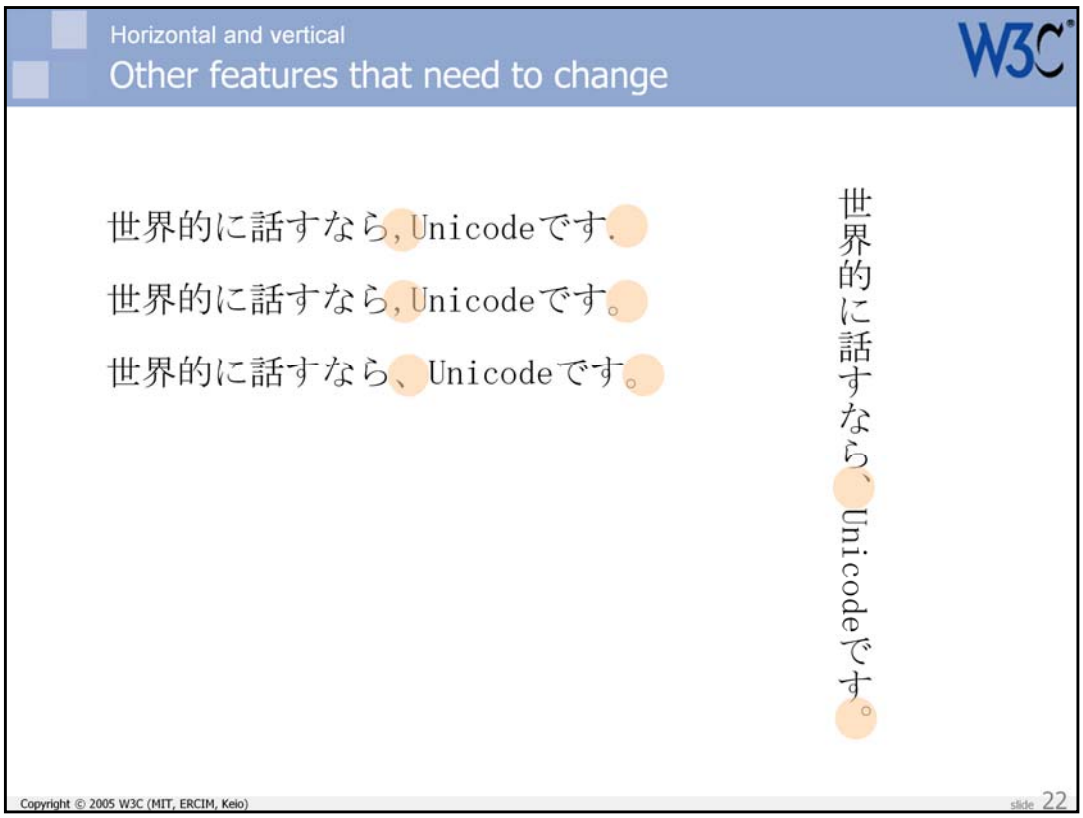
But there are also other differences. Some can be worked around, others are less tractable. Let's take a look at some of the common differences between horizontally and vertically set Japanese text.



A Latin-script acronym or single letter is typically rendered without rotation in vertical text. It also uses a full-width shape, whereas horizontal text will typically use proportionally-spaced characters.

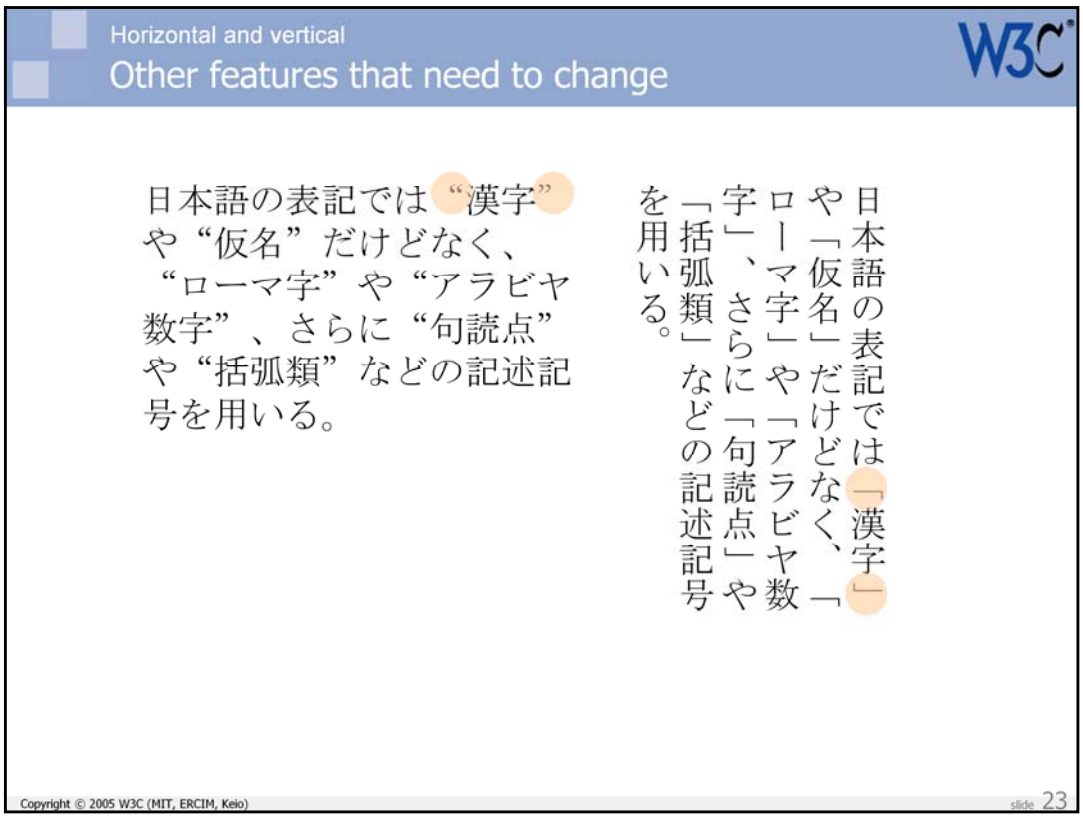
Short two-digit numbers are often horizontally set within the vertical line (this is called tate-chuu-yoko).

In both of these cases, you would need to have markup in the horizontal text to automatically achieve the desired result when the switch to vertical is flipped.



Punctuation characters tend to be used differently in horizontal and vertical text.

Due to Western influence, Japanese authors sometimes use Western commas and punctuation in text. Vertical text, on the other hand, should only use the ideographic comma and full stop.

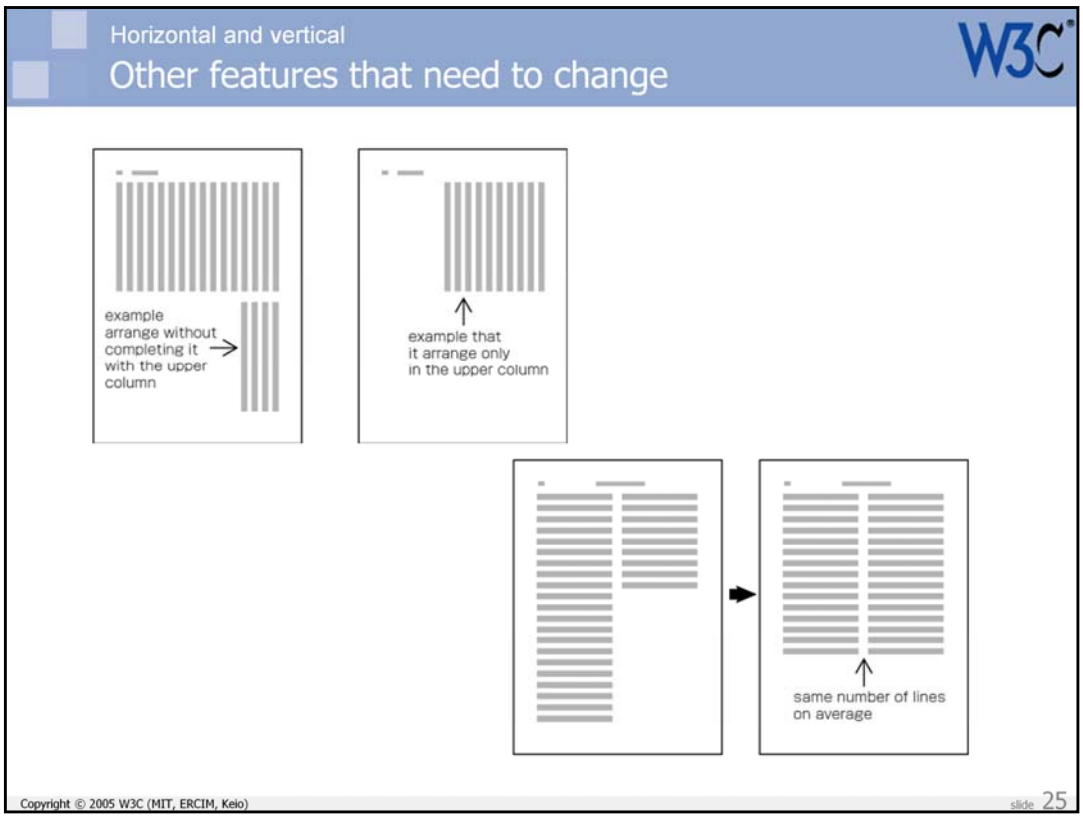


Similarly, there are some differences in the usage of quotation marks in horizontal vs. vertical text. For example, horizontal text often uses double or single quotation marks as an alternative to left and right corner brackets. Vertical text, on the other hand, doesn't use the quotation marks (except sometimes for embedded Latin runs); it uses corner brackets or double prime marks. Double prime marks are not used for horizontal text.

Similar rules apply to parentheses. Left and right tortoise shell brackets (﹝ ﹞) are a vertical equivalent of left and right square bracket ([ ]). Square brackets should be used in horizontal text except for special cases.

Ideographic numerals were traditionally used in vertically set text, instead of Western-Arabic numerals. (Road numbers and car numbers are examples of exceptions). However, newspapers and other publications have been adopting Western-Arabic numerals more in vertical writing mode.

In vertically set text, symbols for units are usually described with katakana characters, eg. センチメートル(centimeter) or センチ(abbreviated form for centimeter). In horizontally set text, the International System of Units(SI) is usually used, eg. "cm".



Where a new page is started and the current page isn't full of text, there may be different strategies for balancing columns on the current page. Typically, text columns in vertically set text end without any balancing. Columns in horizontally set text, however, are typically balanced so that the two columns are the same length.

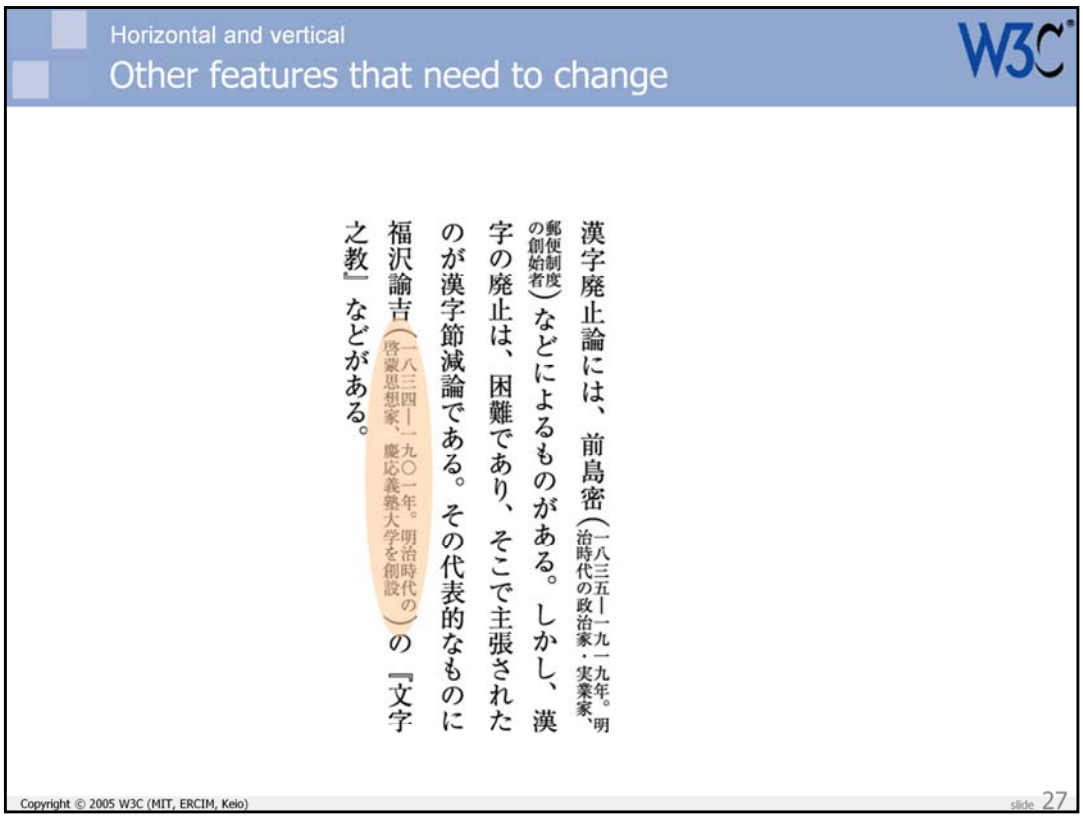
Horizontal and vertical

Other features that need to change

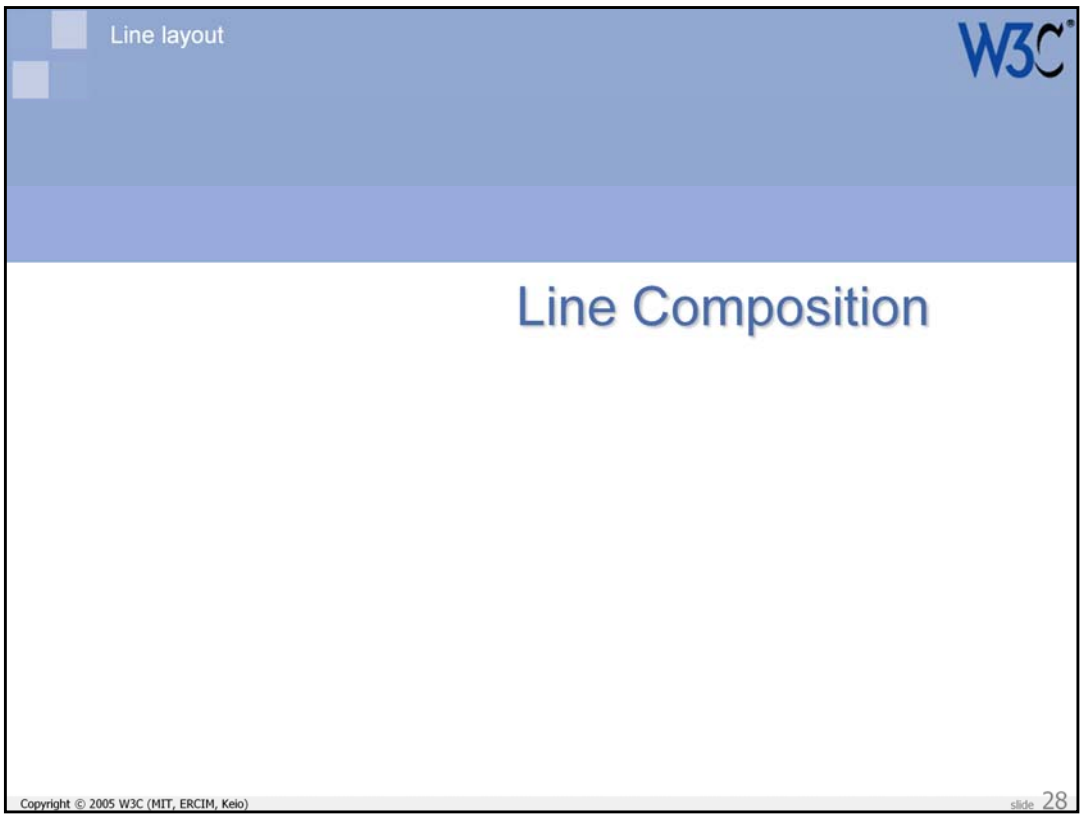
Copyright © 2005 W3C (MIT, ERCIM, Keio)

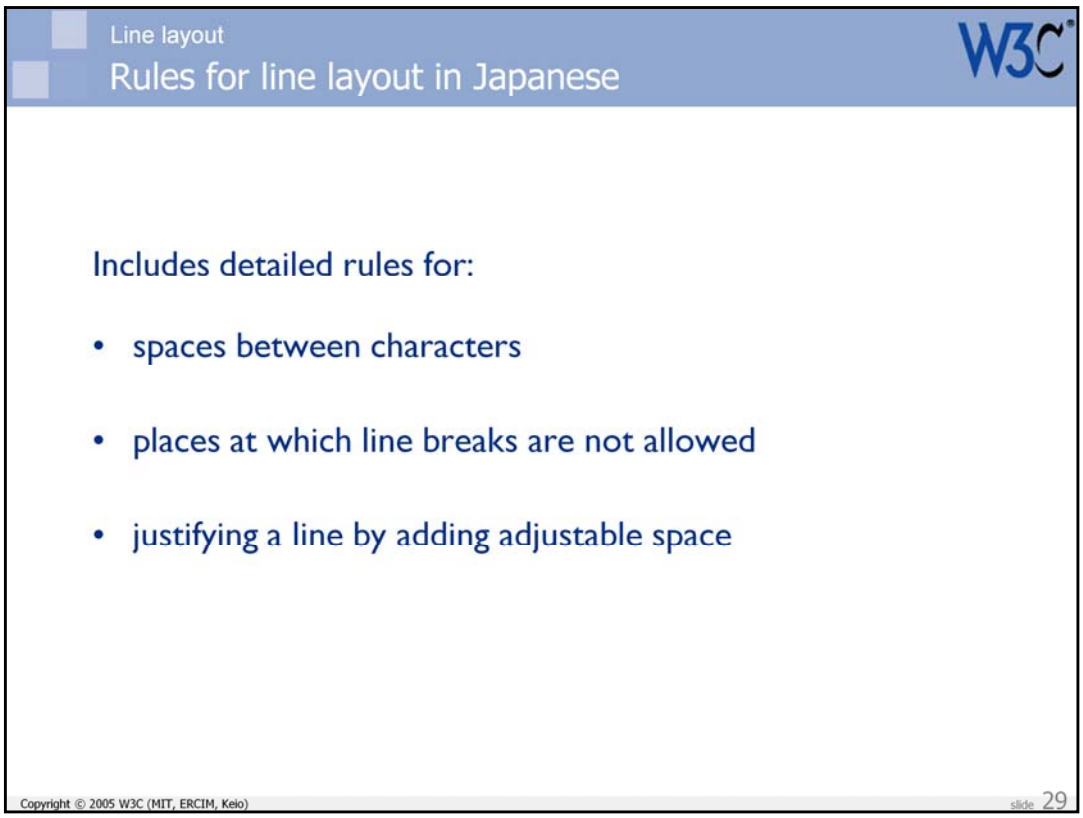
There are also differences in page layout details between horizontally and vertically set pages that are not automatically achieved.

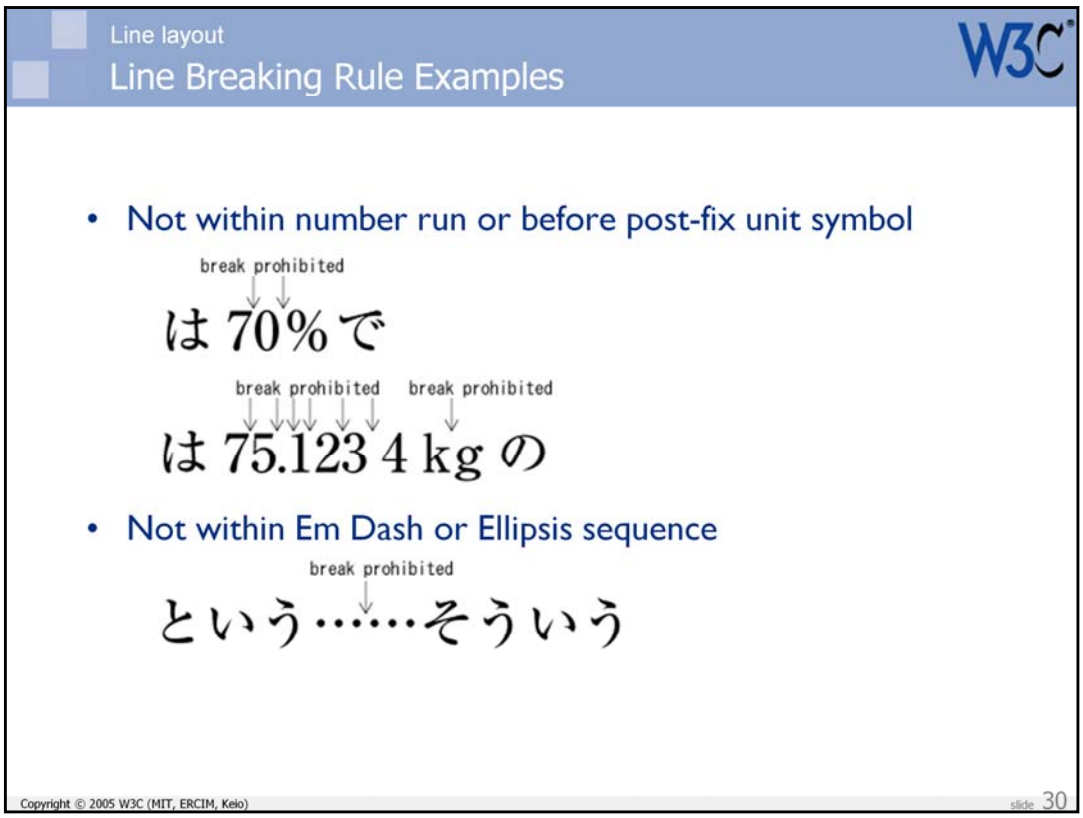
For example, when positioning running heads and page numbers horizontally with reference to kihon hanmen in vertical writing mode, the amount of vertical space between the edge of kihon hanmen and the running head is one full width character size based on the kihon hanmen character sizes. If kihon hanmen is horizontally set, you need to take more vertical space than the character size in kihon hanmen.



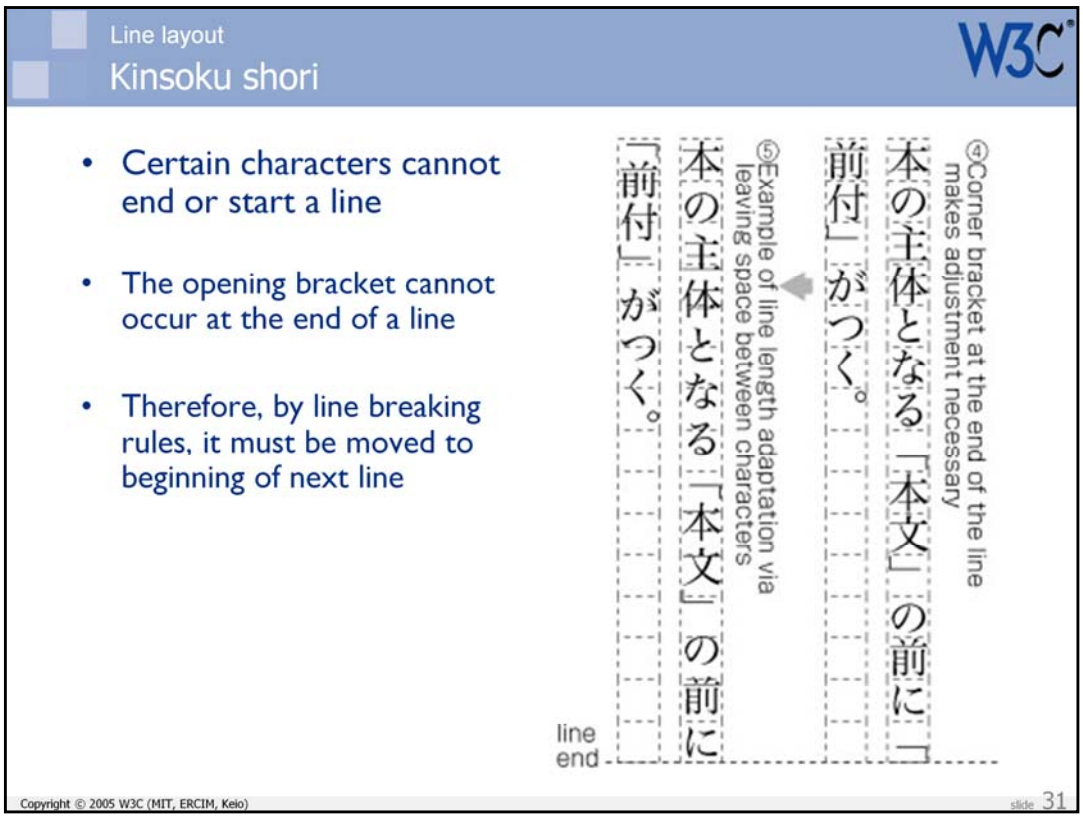
Typographic conventions such as warichu are much more common in vertically set text than horizontal.


Line layout
Line Composition
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 28

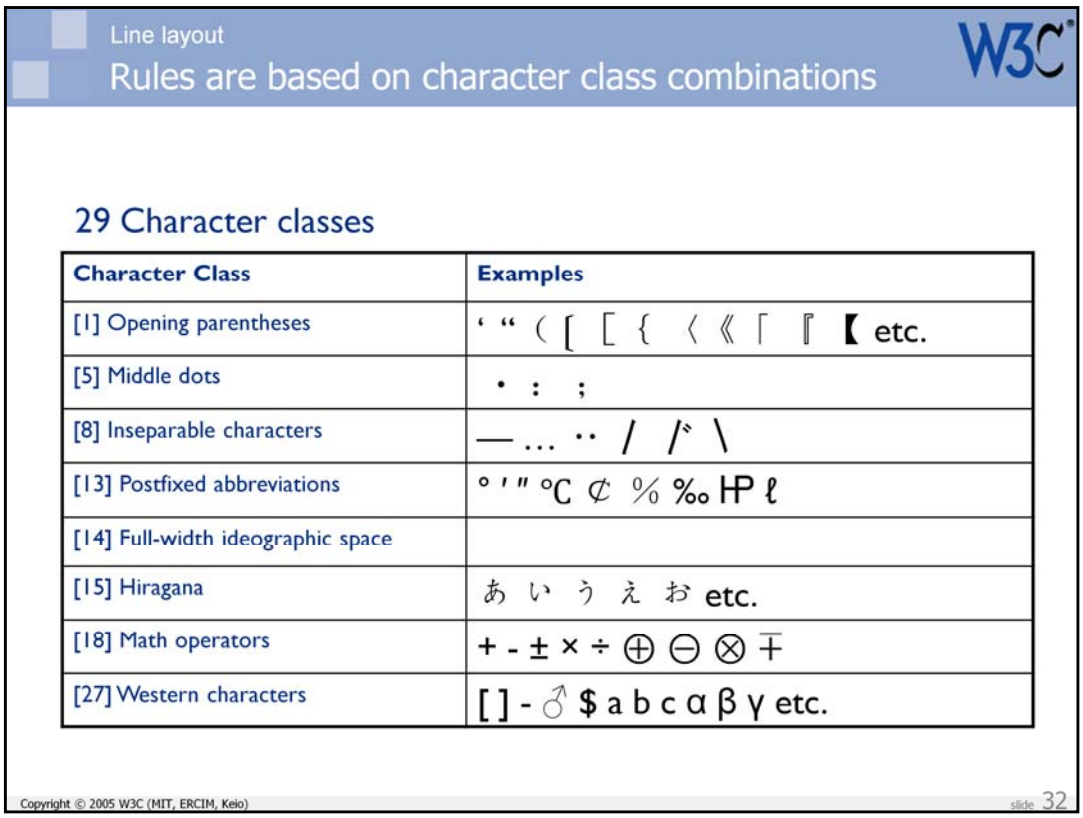

Line layout
Rules for line layout in Japanese
Includes detailed rules for:
• spaces between characters
• places at which line breaks are not allowed
• justifying a line by adding adjustable space
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 29



There are rules about line breaking that forbid line breaks in certain contexts.



In addition, certain characters cannot start a line, and others cannot end a line according to the layout rules. Action has to be taken to deal with situations where the normal progression of characters would lead to such a character appearing in an illegal position.



Line layout

Rules are based on character class combinations

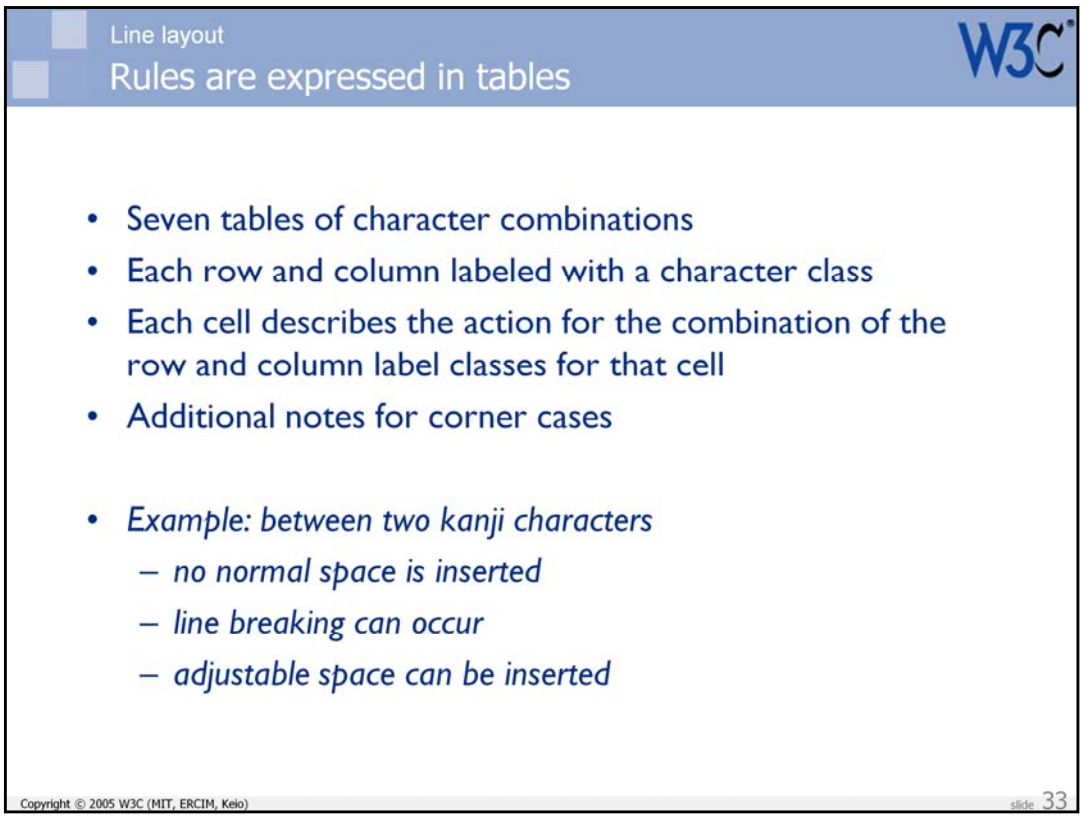
29 Character classes

Character Class	Examples
[1] Opening parentheses	‘ “ ([［ { 〈 《 「 『 【 etc.
[5] Middle dots	・ : ;
[8] Inseparable characters	— … ‥ / ゙ \
[13] Postfixed abbreviations	° ′ ″ ℃ ¢ % ‰ ㍻ ℓ
[14] Full-width ideographic space	
[15] Hiragana	あ い う え お etc.
[18] Math operators	+ - ± × ÷ ⊕ ⊖ ⊗ ∓
[27] Western characters	[ ] - ♂ $ a b c α β γ etc.

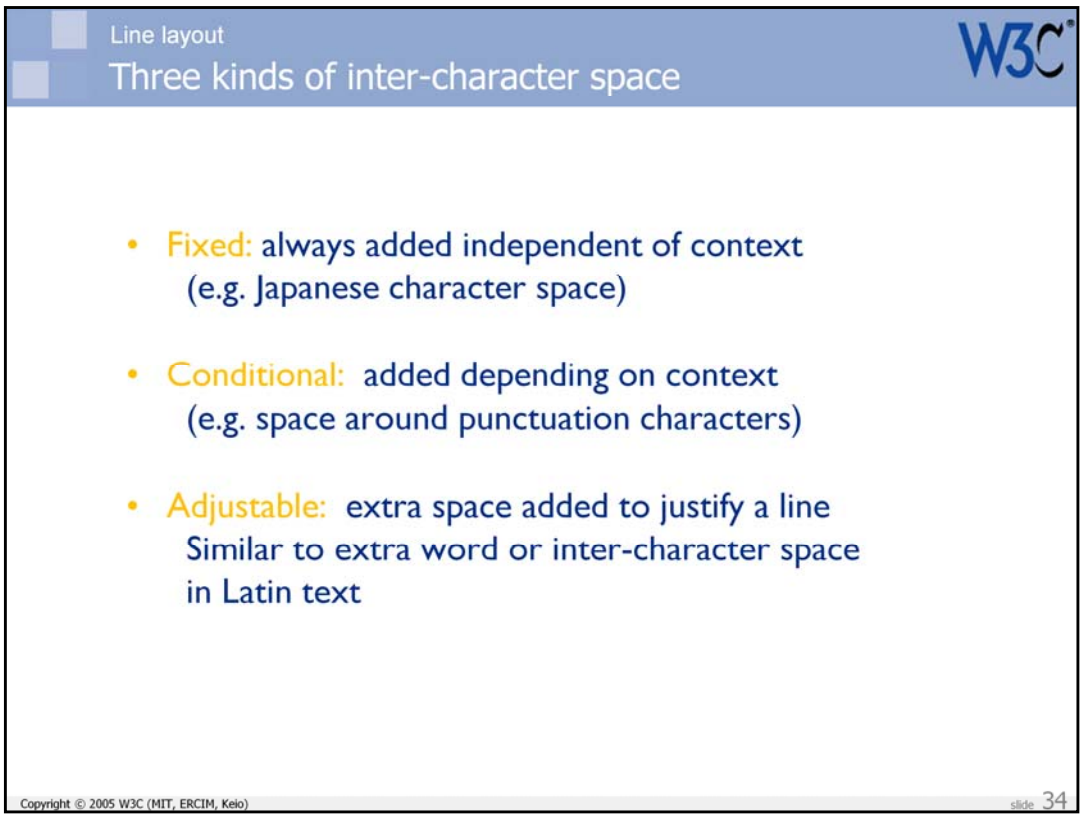
Copyright © 2005 W3C (MIT, ERCIM, Keio)

slide 32

In the document Requirements for Japanese Layout, the rules for such behavior are specified by assigning characters to one of 29 classes. Characters within a given class exhibit similar behaviors.



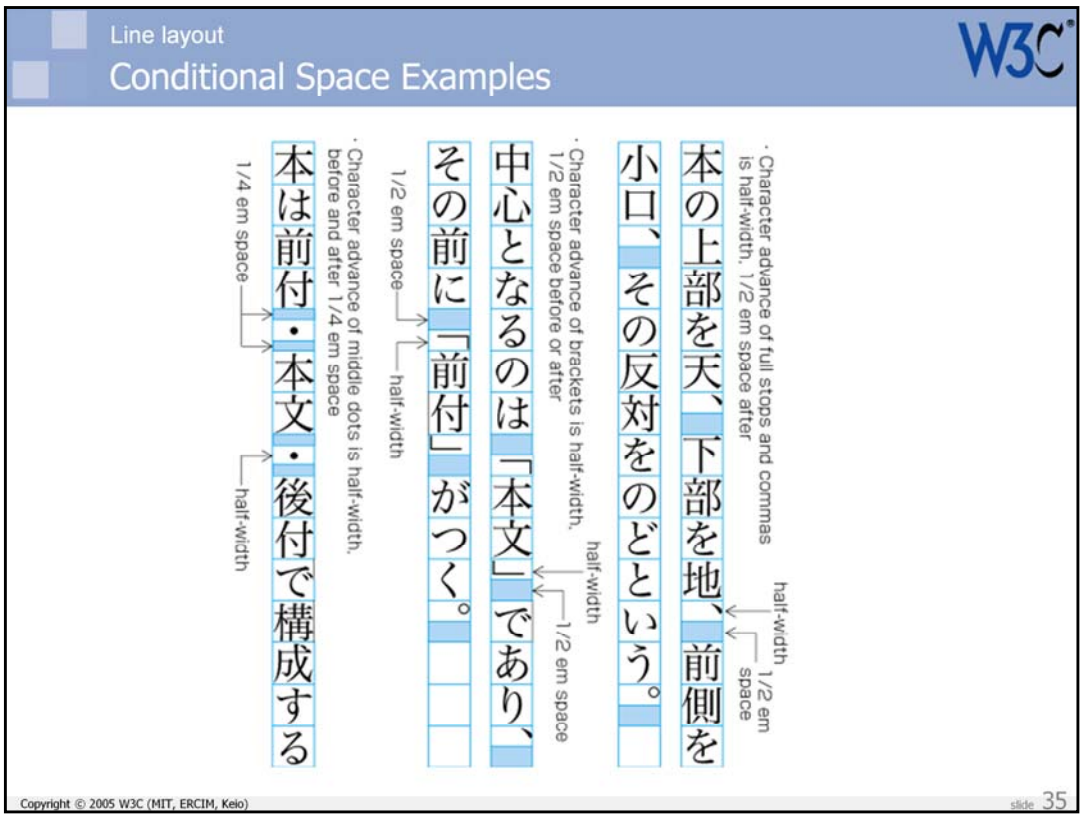
The document then contains tables which indicate the expected behaviors for given combinations of characters. There are additional notes for corner cases.



One important set of information in the document concerns the handling of visual space in Japanese text. Space can be created using explicit space characters (ie. fixed spacing), but this is very rare.

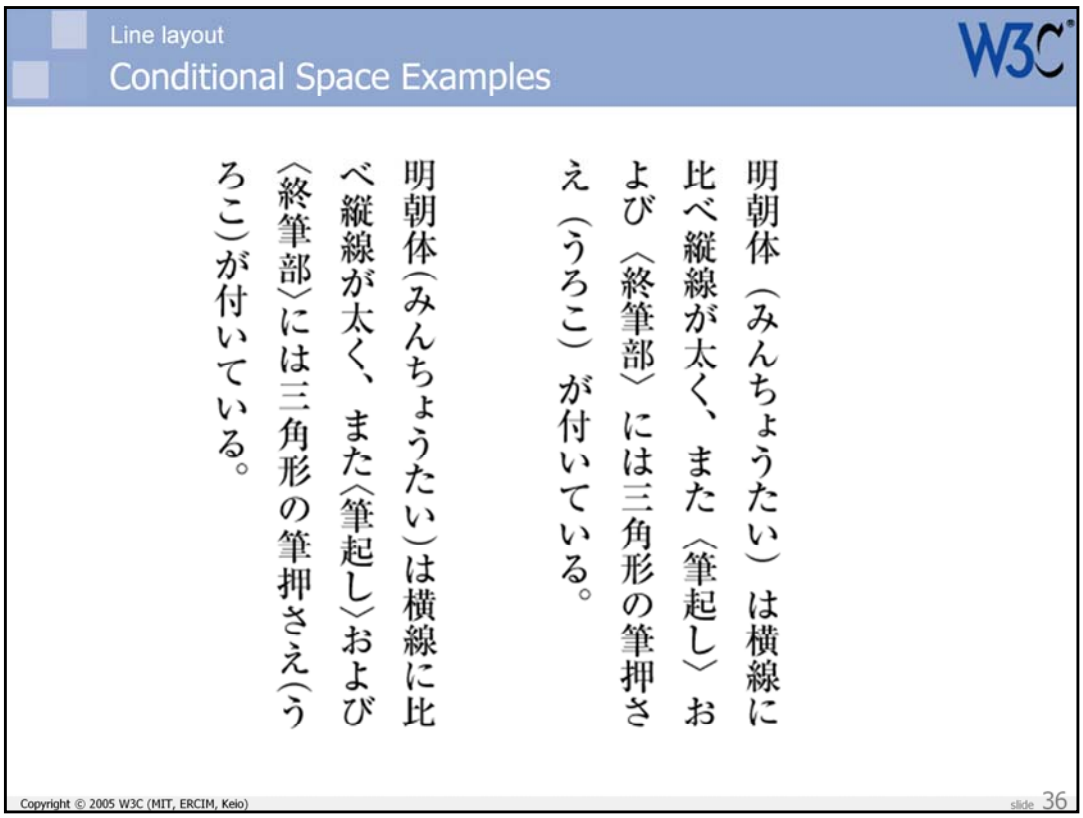
On the other hand, there are rules for the introduction or elimination of visual space between characters when they are rendered. The starting point for this is to understand that many of the punctuation marks used in Japanese are considered to intrinsically be only half the size of the normal kanji, hiragana or katakana character. The document then specifies conditional spacing rules to indicate how visual space should be added to the displayed text to produce the best effect.

We have also seen a number of situations that tend to break the regularity in the placement of characters in a line in the kihon-hanmen. These include rules forbidding line breaks in particular sequences of characters, and rules about which characters can appear at the beginning or end of a line. In such cases, adjusting a line so that the last character is always flush with the end of the other lines involves tweaking the spaces between characters on a line. This is done by looking for certain character combinations first, and tweaking those, then looking for others if that was not enough. The document also specifies rules for this adjustable spacing.

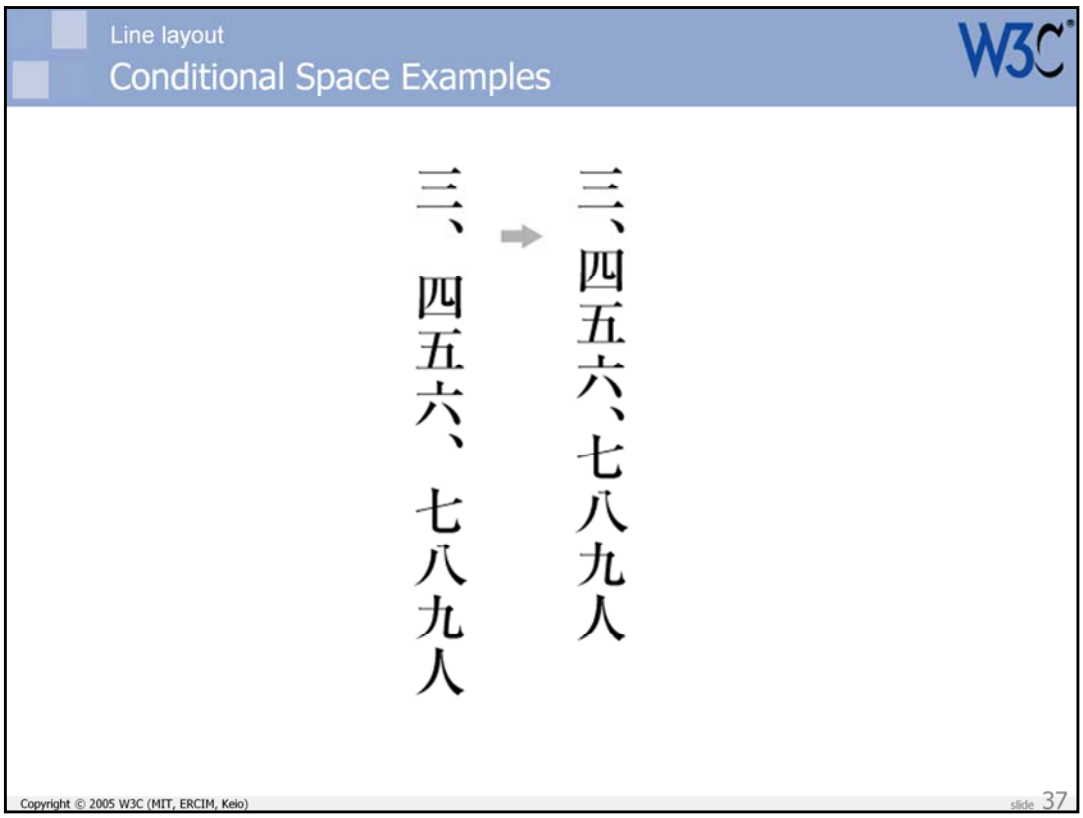


In these examples of conditional space rules, you can see that closing parentheses, and ideographic commas and full stops are usually followed by a half-width space, and opening parentheses are preceded by a half-width space. Middle dot characters are usually surrounded by quarter-width spaces. This maintains the appearance of equal character advances for the characters.

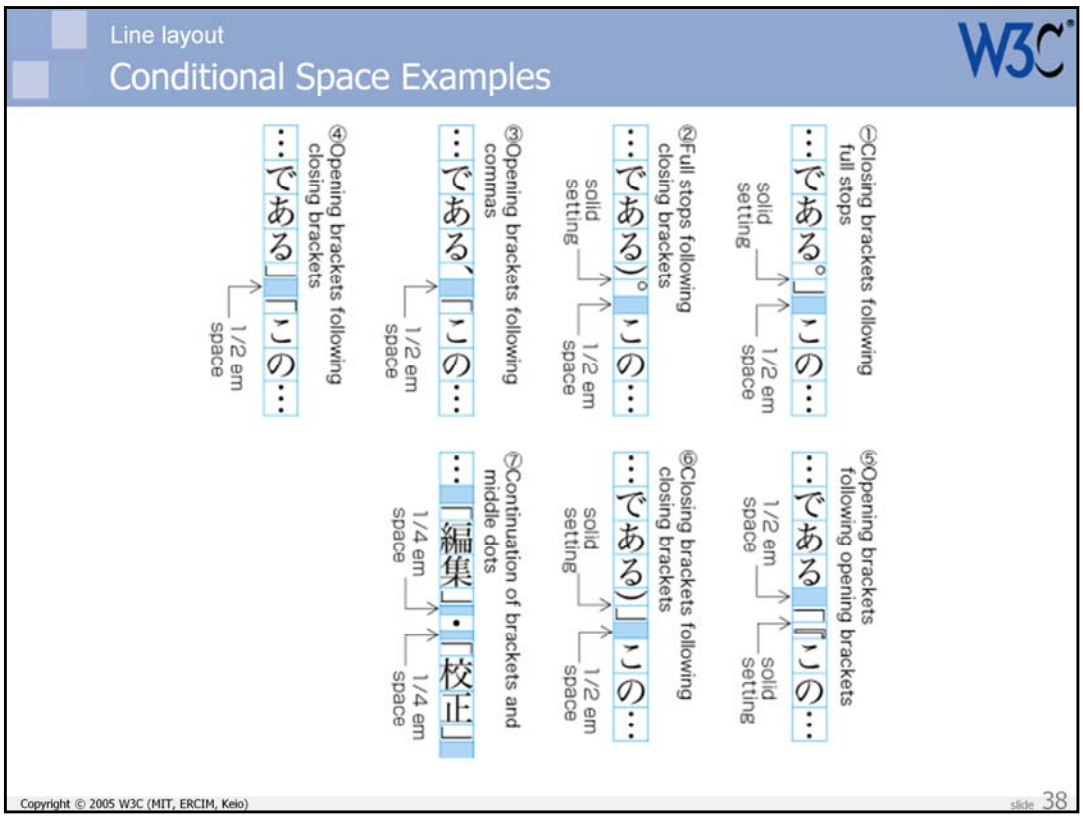
These kinds of rules are expressed in the document.



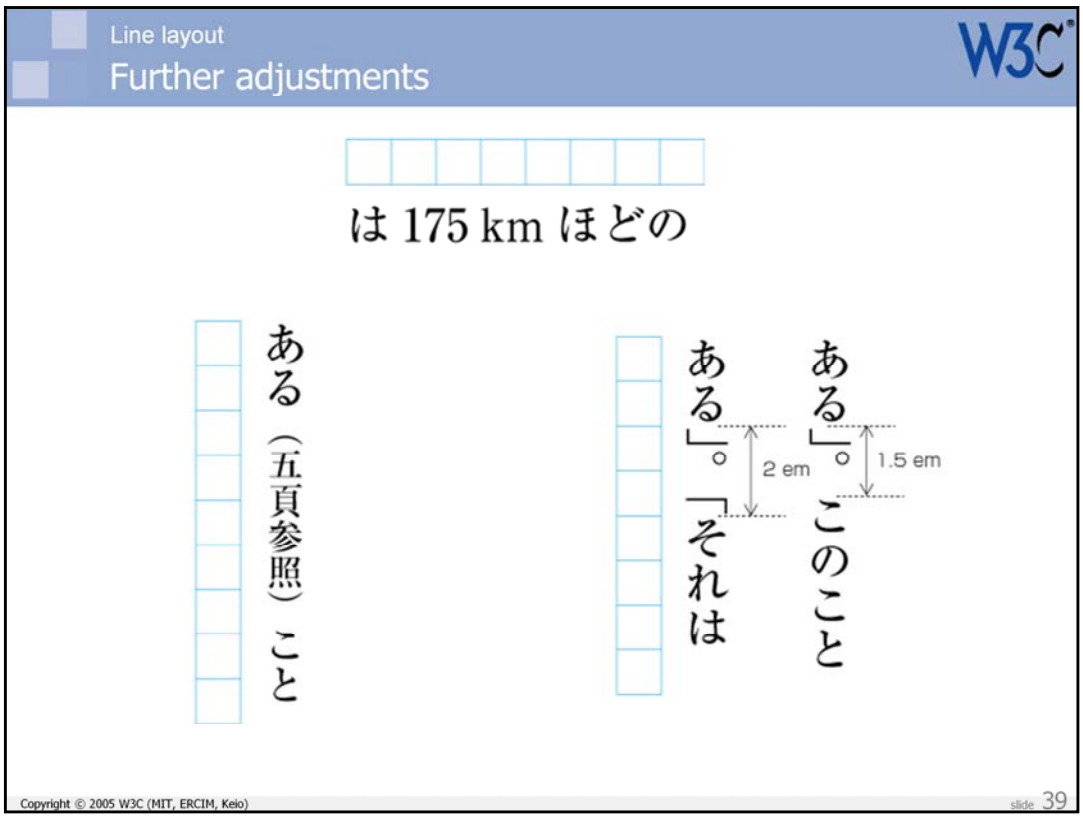
In some cases, the rules are slightly different, according to the context. For example, where parentheses contain information that clarifies the preceding text, sometimes the preference is to remove the spacing associated with the parens, to make the explanation closer to the other text.



Another exception arises where the role of the character is different. In the example shown here, the comma is part of the number, and so the space is removed.

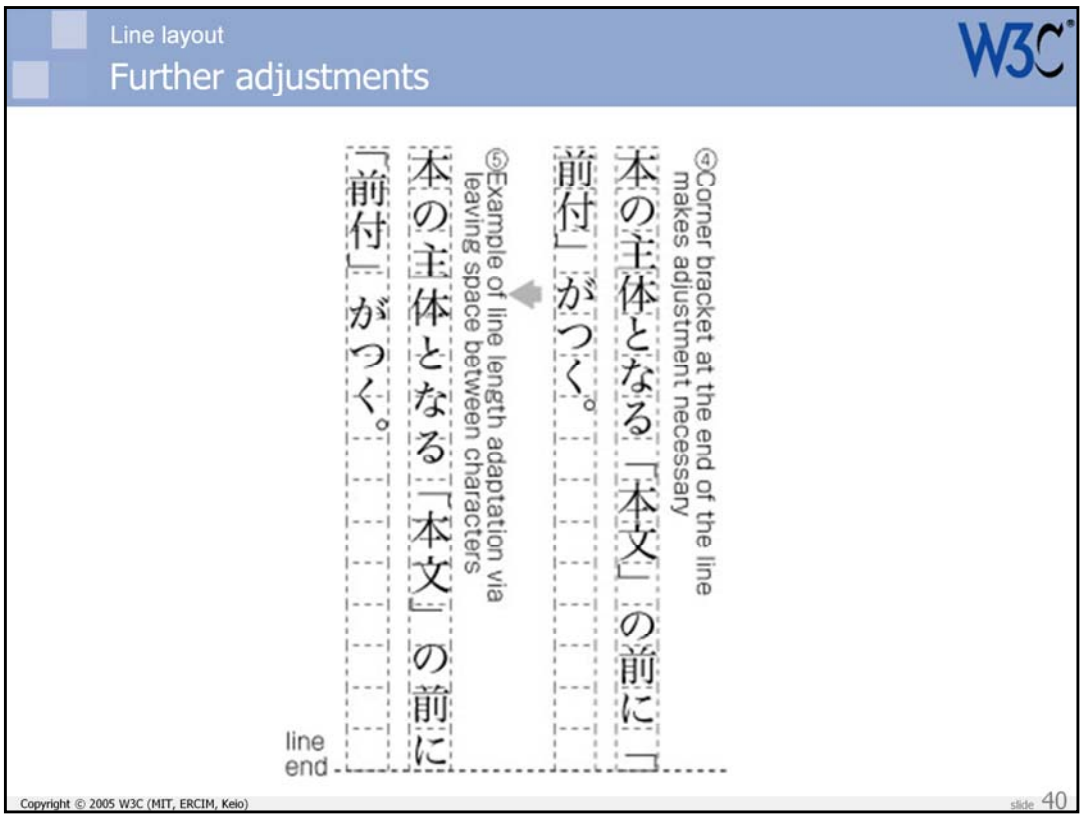


However, if a closing and opening parenthesis are side-by-side, or a comma/period is next to a parenthesis, only one half-width space is needed between them. (This may lead to the need to adjust spacing later.)



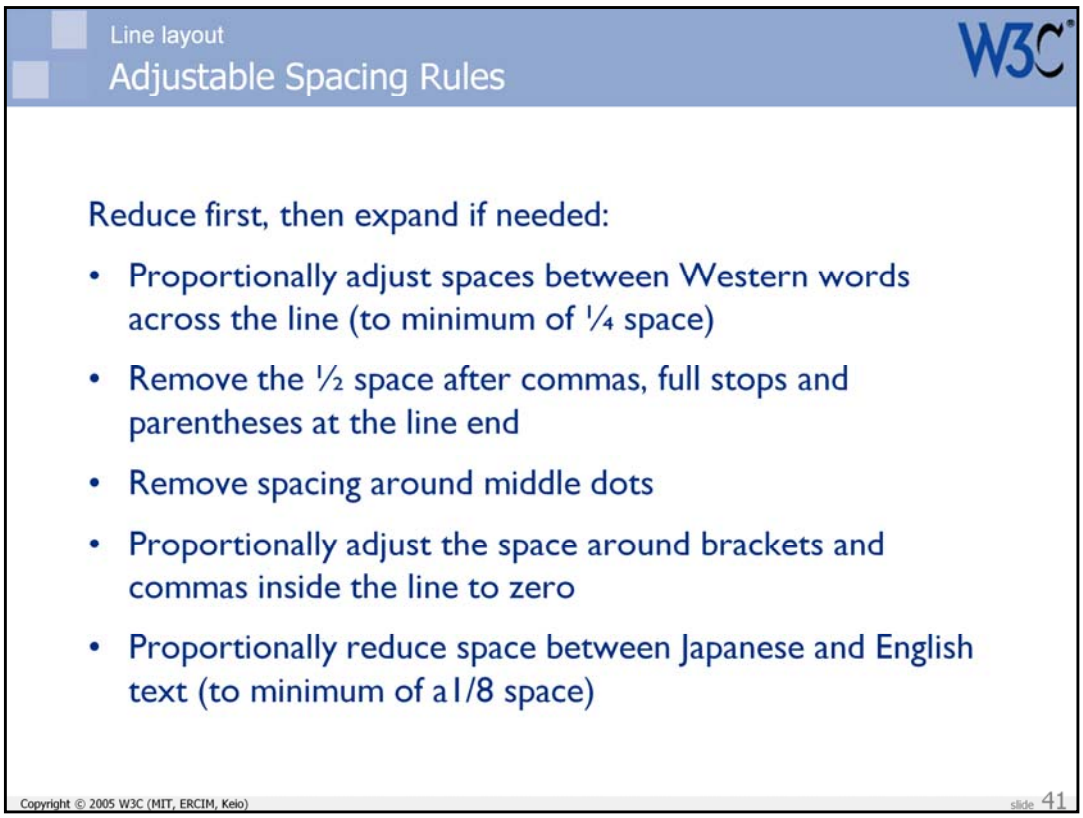
These are further examples of situations where the regular progression of characters, one after the other, would result in problems at the end of the line. They include non-kanji/kana text, smaller text sizes, and multiple punctuation characters side-by-side.

In these cases, we will need to tweak the spacing between characters on the line to ensure that the last character on the line is flush with the line end.

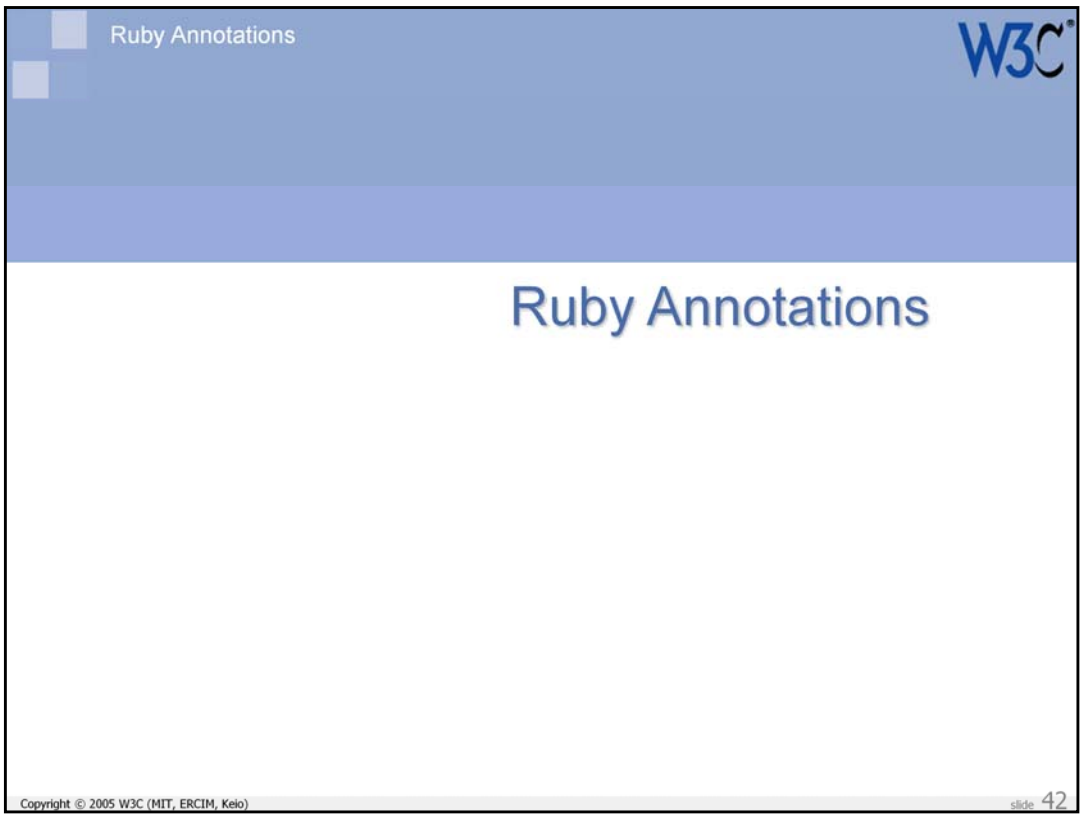


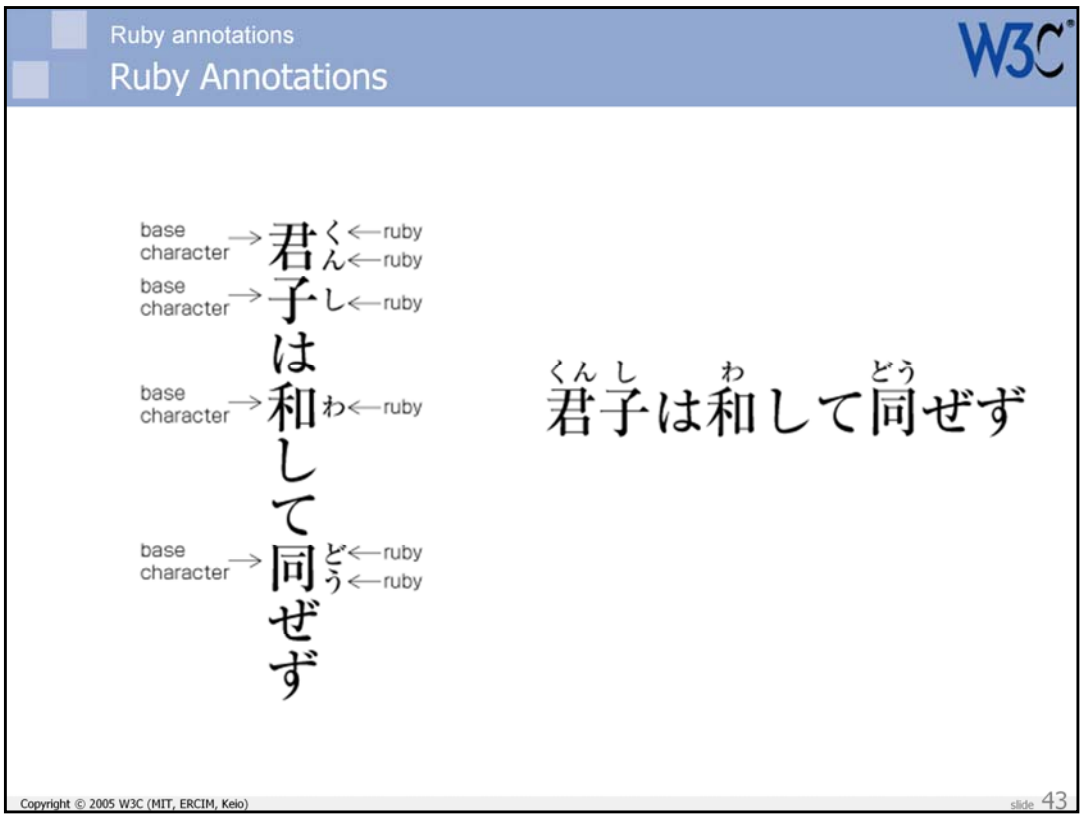
The approach involves targeting specific character combinations, in order, and changing the virtual space associated with them by small amounts, according to rules.

An attempt is made to solve the problem by reducing spacing. If that is insufficient, spaces are then expanded.



These illustrate the type of rules involved. Each step is taken, one at a time, until the problem is solved.

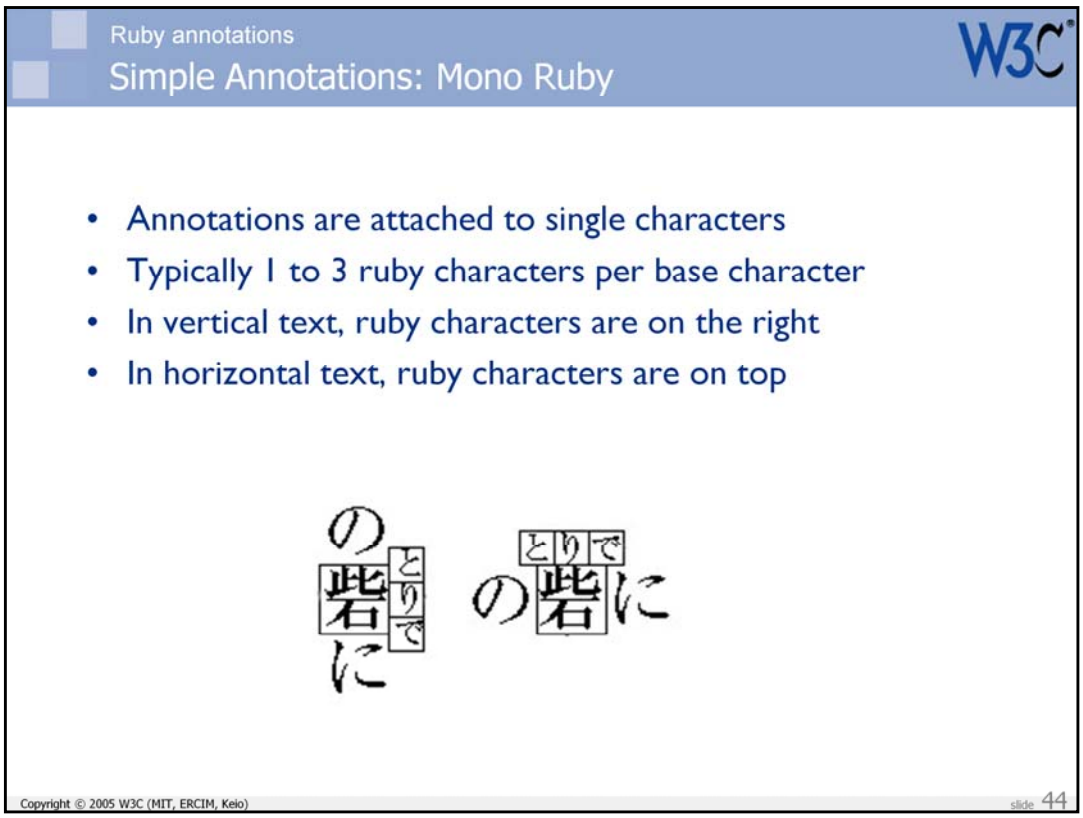

Ruby Annotations
W3C
Ruby Annotations
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 42



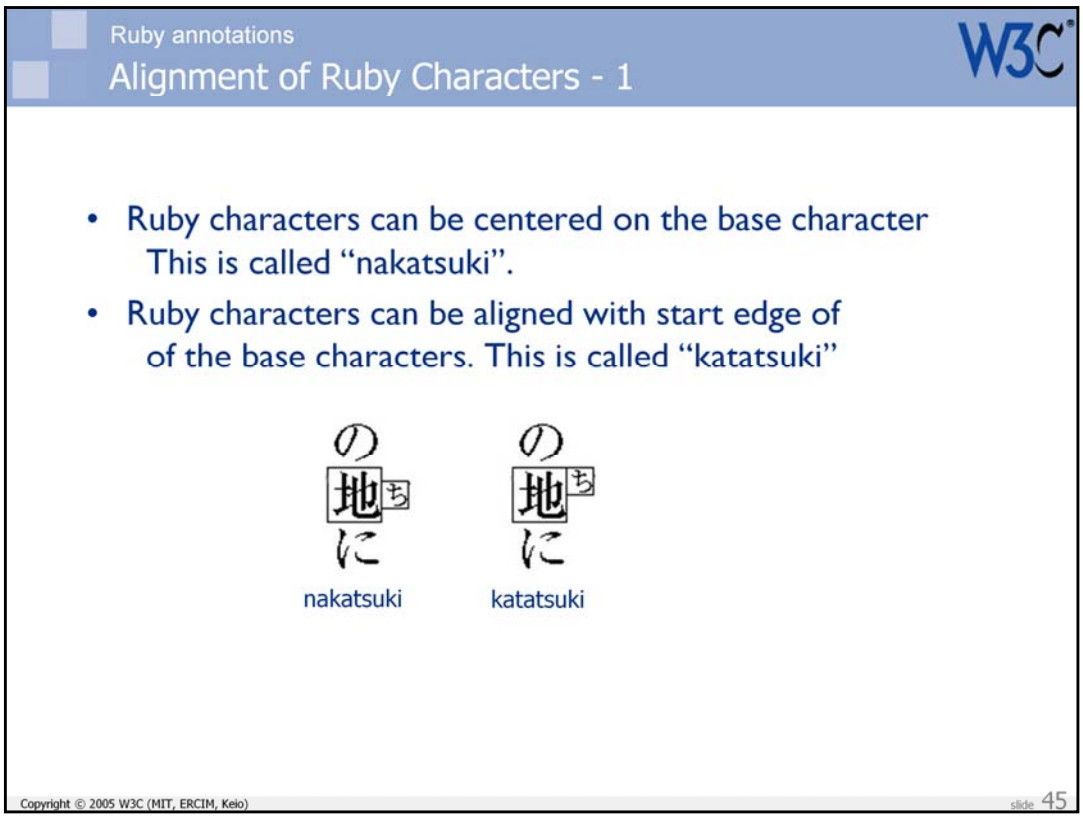
Because you cannot necessarily tell the pronunciation (and sometimes the meaning) of a Japanese character just by looking at it, the Japanese script makes regular use of annotations to help the reader. These are called 'ruby' (a term originally derived from a English typesetting tradition).

Most commonly, ruby appears to the right of vertical text and above horizontal text.

The document describes three main types of ruby: mono ruby, group ruby, and jukugo ruby. We will describe these here.



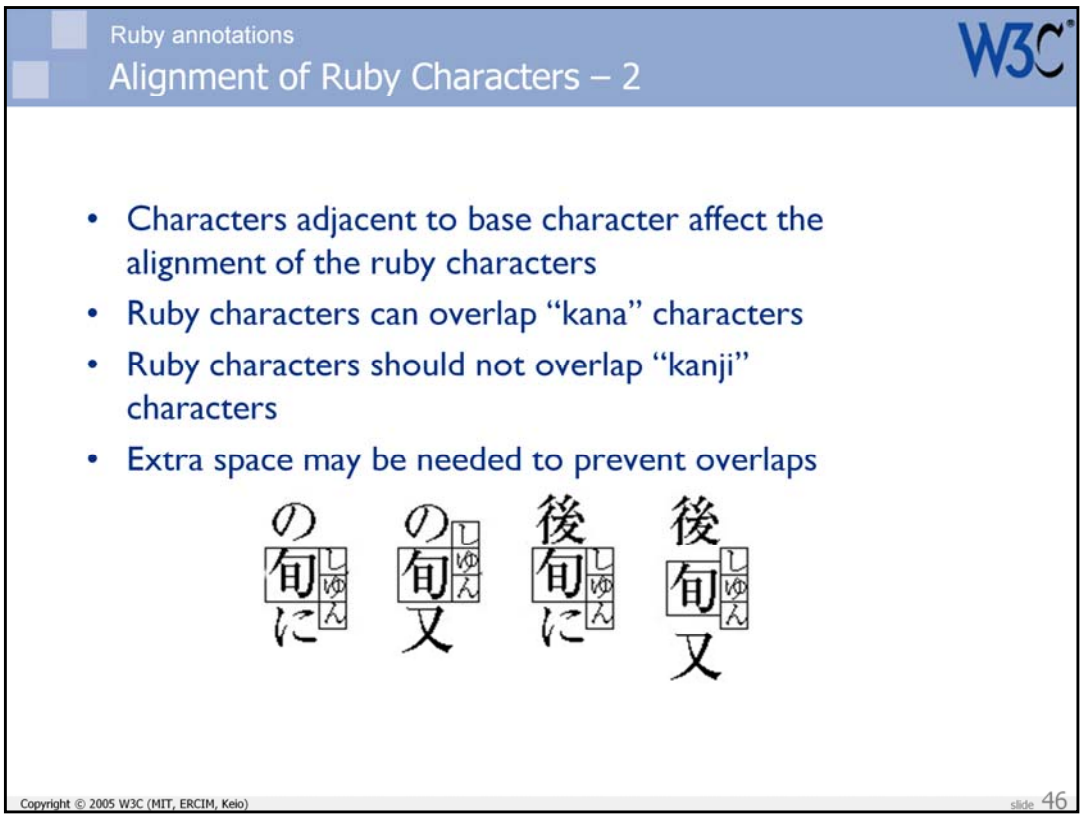
The imaginary box (in which the character is centered) is drawn to more clearly show the positioning of the ruby characters relative to the base character.



There are two approaches to positioning the ruby characters with respect to the base character. If there is an odd number of characters, as is the case here, then the center character can be aligned with the center of the base characters. This is called, "nakatsuki" alignment.

Alternatively, the initial character of the sequence of ruby characters can be aligned with the start edge of the base characters. This is called, "katatsuki' alignment. This approach was more often used with (hot) metal type setting and with digital text layout, the nakatsuki approach is more popular.

Note that if there are two ruby characters attached to the base character, then both approaches give the same results as long as the size of the ruby characters are ½ the size of the base characters.



The ruby annotation is: "shi' "yu" "n" (?)

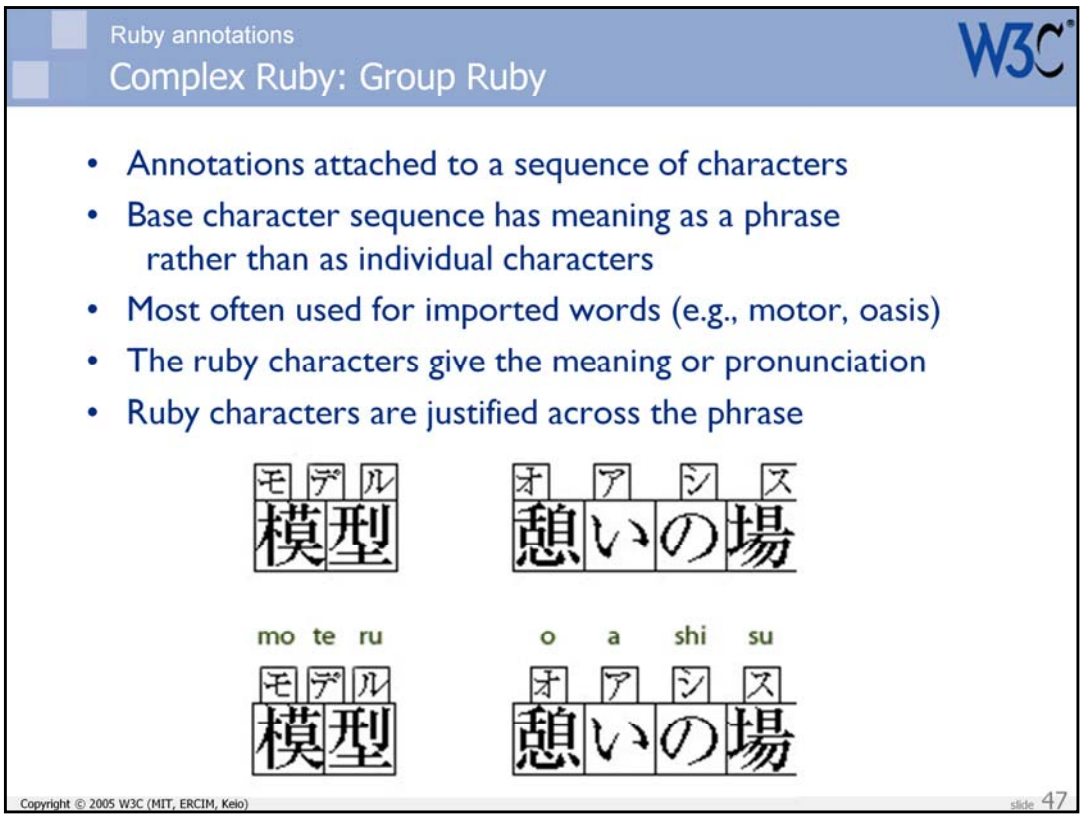
Going from left to right:

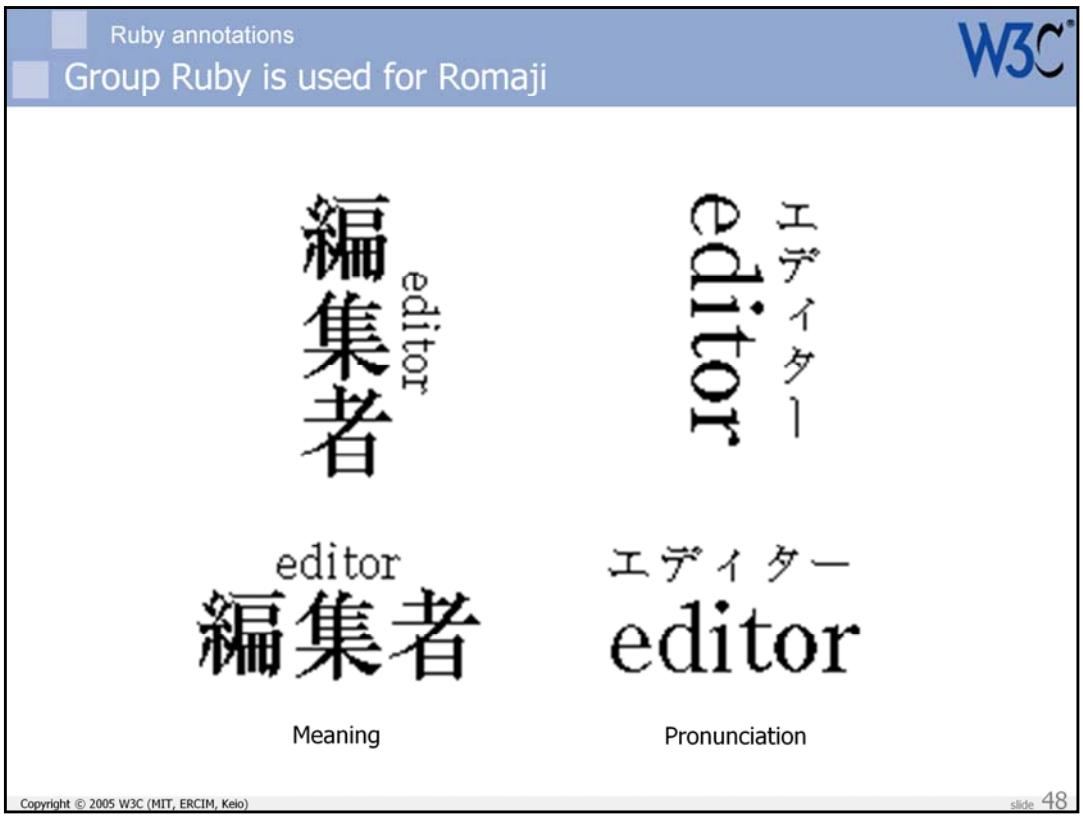
The first sequence has kana before and after the annotated kanji so overlap is allowed on either side, but the in katatsuki the first ruby character is normally aligned with the start edge of the base characters

The second sequence as a kanji character (hand) following the annotated kanji so the only allowable overlap is with the hiragana character that is before the annotated base character

The third sequence has a kanji before the annotated base character and kana afterwords, so the normal katatsuki rule works (actually is forced) in this case.

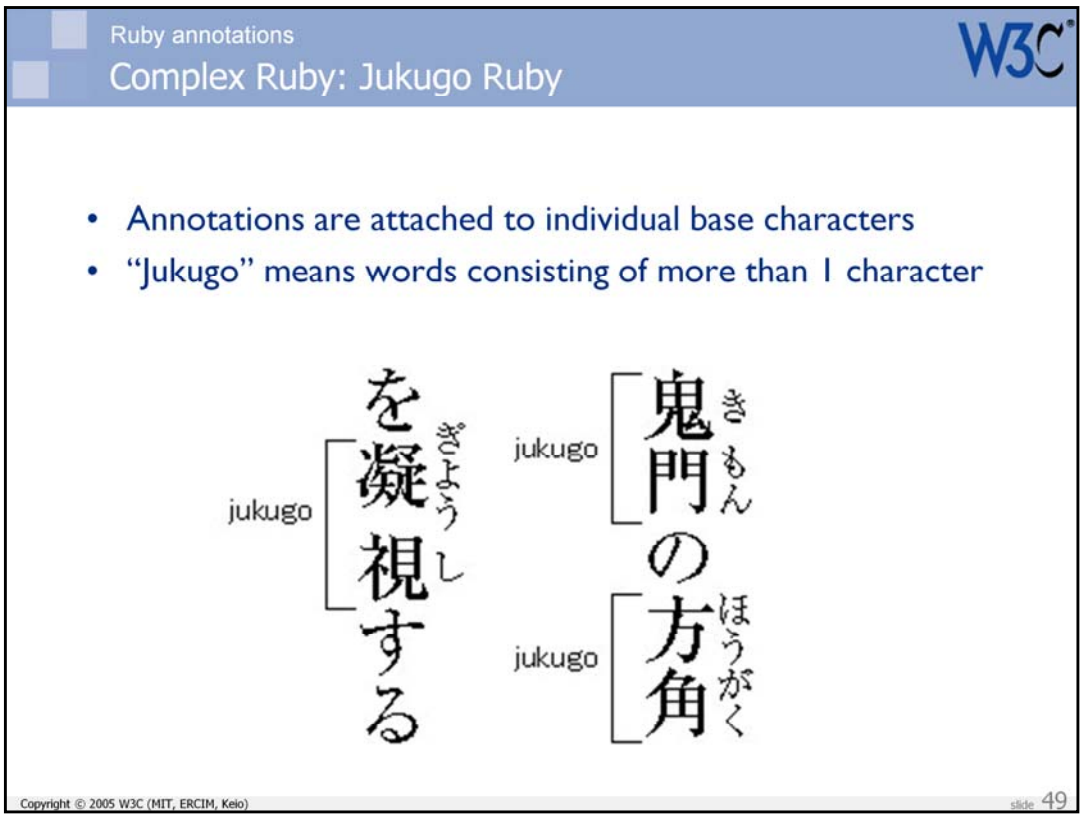
In the fourth and final sequence, there are kanji characters before and after the annotated base character and, therefore, it is necessary to (a) use nakatsuki alignment and (b) introduce space between the annotated base character and the kanji charaacters on either side. Using nakatsuki alignment means that the extra space is distributed equally on both sides of the annotated base character.


Ruby annotations
Complex Ruby: Group Ruby
W3C
• Annotations attached to a sequence of characters
• Base character sequence has meaning as a phrase rather than as individual characters
• Most often used for imported words (e.g., motor, oasis)
• The ruby characters give the meaning or pronunciation
• Ruby characters are justified across the phrase
モデル
模型
オアシス
憩いの場
mo te ru
o a shi su
モデル
模型
オアシス
憩いの場
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 47



In the lefthand set, the kanji is given an English meaning, "editor". Note that the latin characters are formatted/justified as the normally would be rather than spreading them across the whole "width" of the kanji base character sequence.

In the righthand set, the pronunciation of the English word, "editor" is given in katakana, which is evenly distributed across the "width" of the base (romanji) characters.



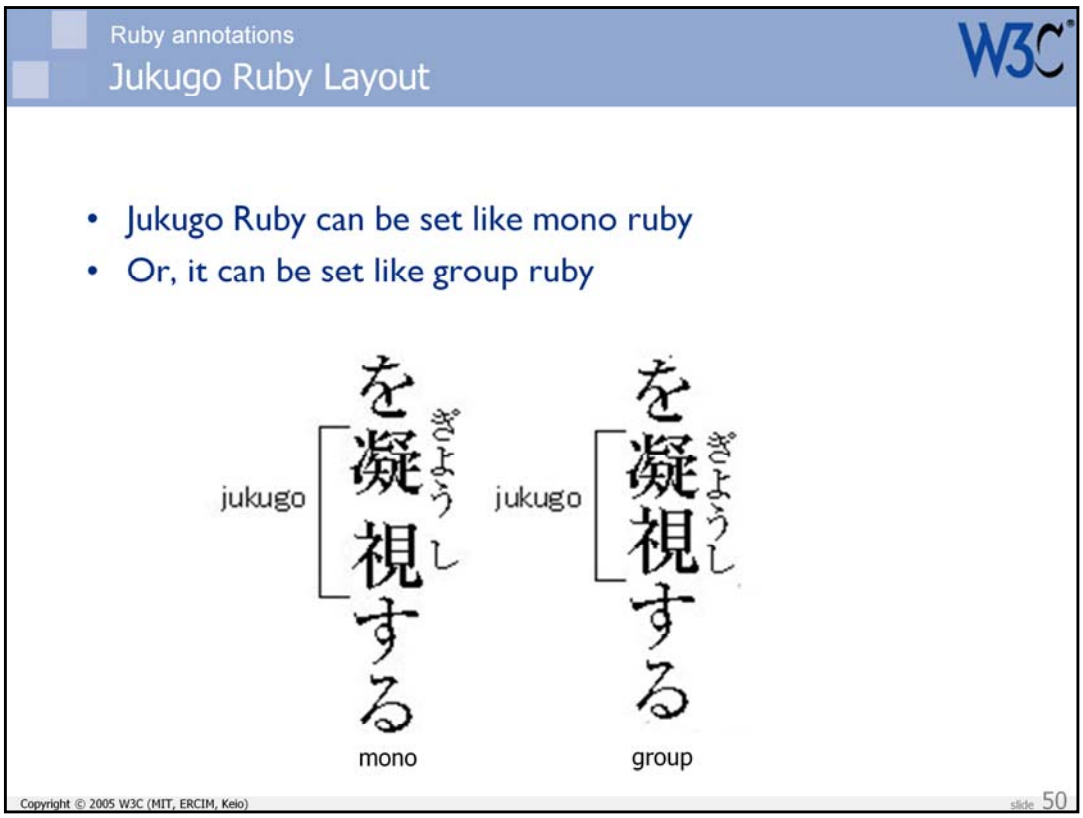
The example here is in two lines of Japanese and is read from right to left vertically.

Jukugo refers to compound words in Japanese, which are typically composed of two or three kanji characters.

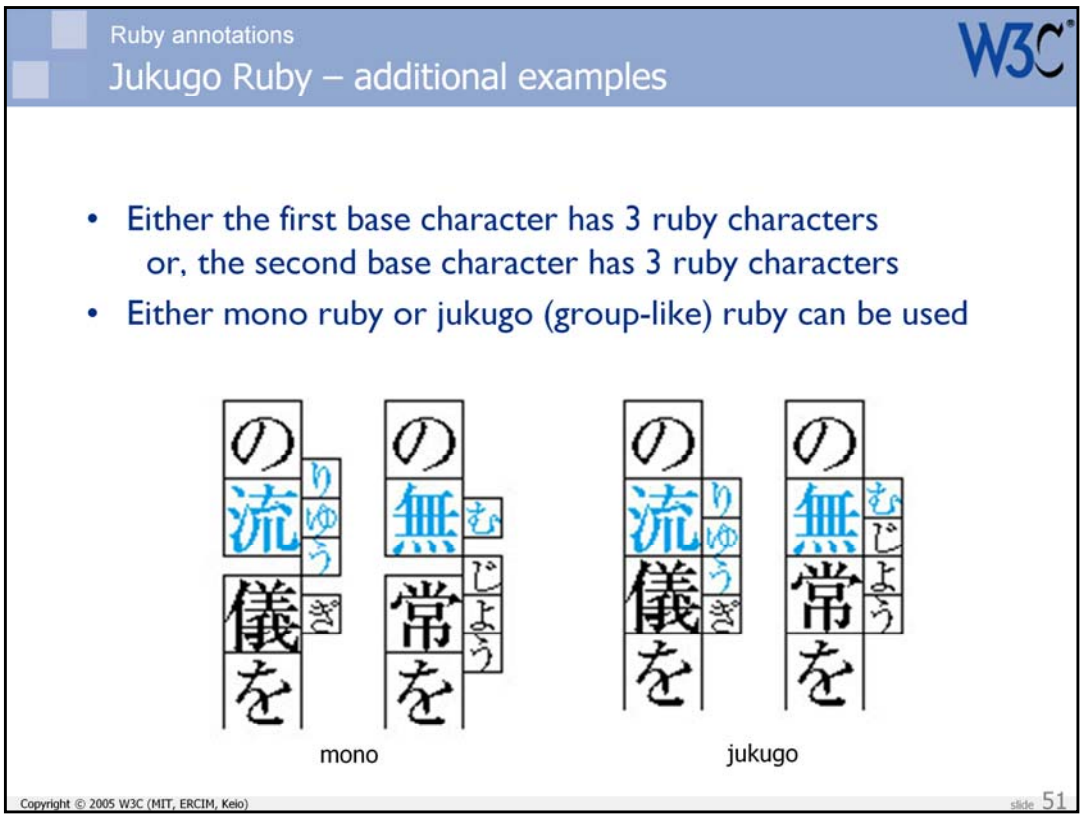
The first jukugo, the one on the top right, has only three ruby characters in its annotation: one on the first base charaacter and two on the second base character

The second jukugo, the one on the bottom right, has two ruby characaters on each of its two base characters

The third jukugo, the one on the left, has three ruby characters on its first base character,but only one ruby character on its second base character. Because the ruby cannot overlap a kanji character, there must be a space between the first and second kanji characters. But, this is somewhat ugly. so, next slide



In this slide, we have repeated the mono ruby setting of jukugo ruby, but show that a more beautiful setting results from treating the jukugo ruby as if it were group ruby.



Here are two examples of jukugo text with attached ruby annotations.

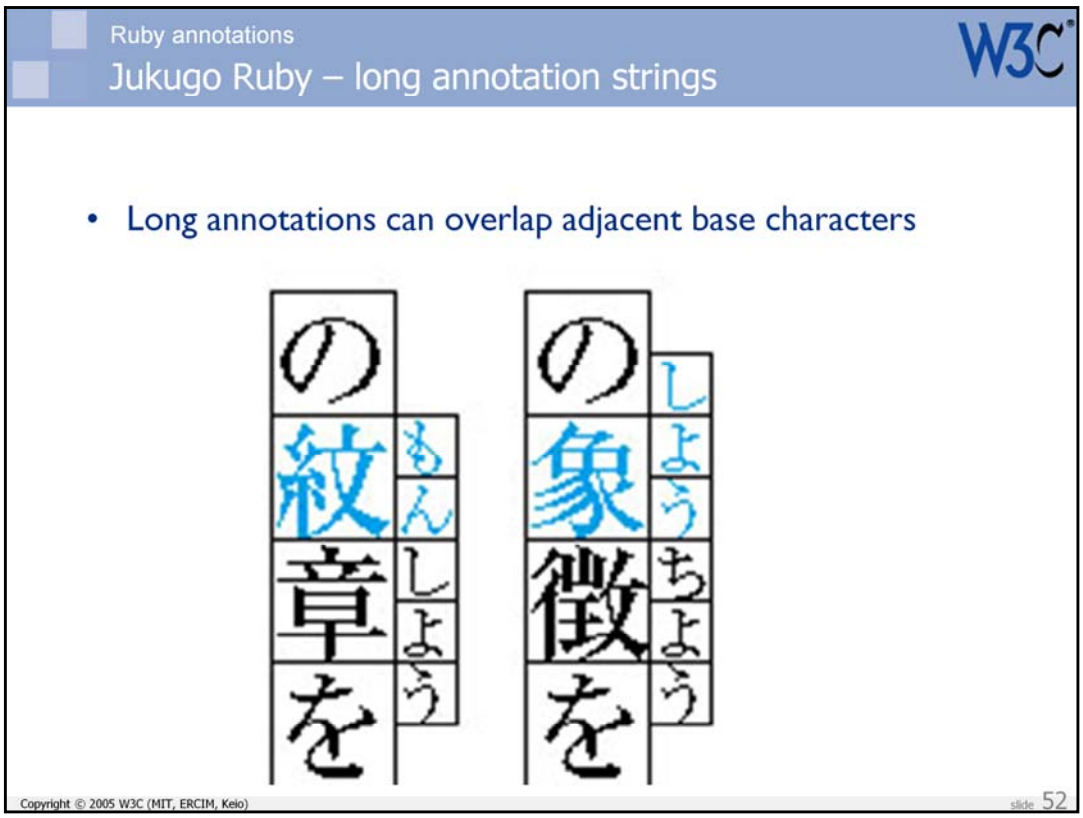
In the first example, the left one, the first base character has three attached ruby characters and the second base character only has one attached ruby character.

In the second example, the right one, the first base character has only one attached ruby character and the second base character only has three attached ruby character

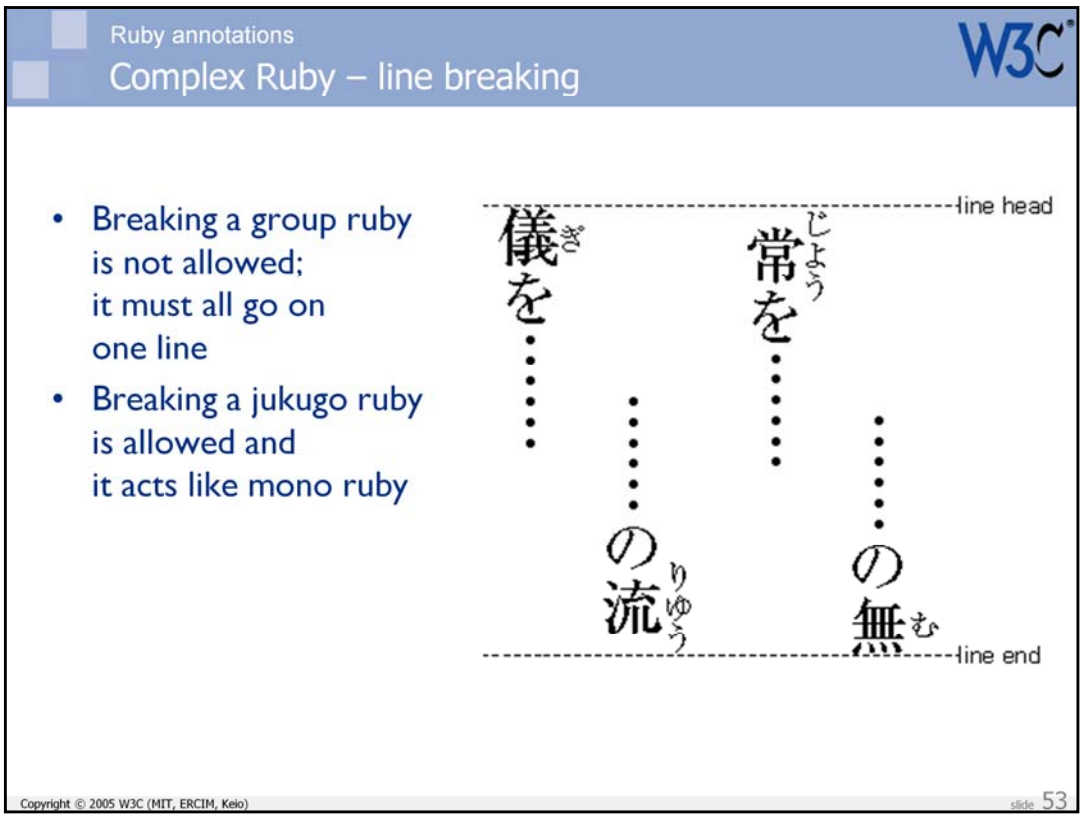
In both cases, the first base character and its attached ruby charaacters are shown in blue and the second base characters of the jukugo and its attached ruby characters are shown in black.

The pair of examples on the left show a setting of both examles using the rules for mono ruby. The second pair of examples show that the rules for jukugo ruby allow a much more pleasing setting of the text. The jukugo setting avoids breaking the Kihonhanmen grid. Note that it does not matter which character has 3 ruby characters attached as long as the other character has only one ruby attached.

So, what happens if the jukugo base characters have more that a total of 4 ruby characters attached?

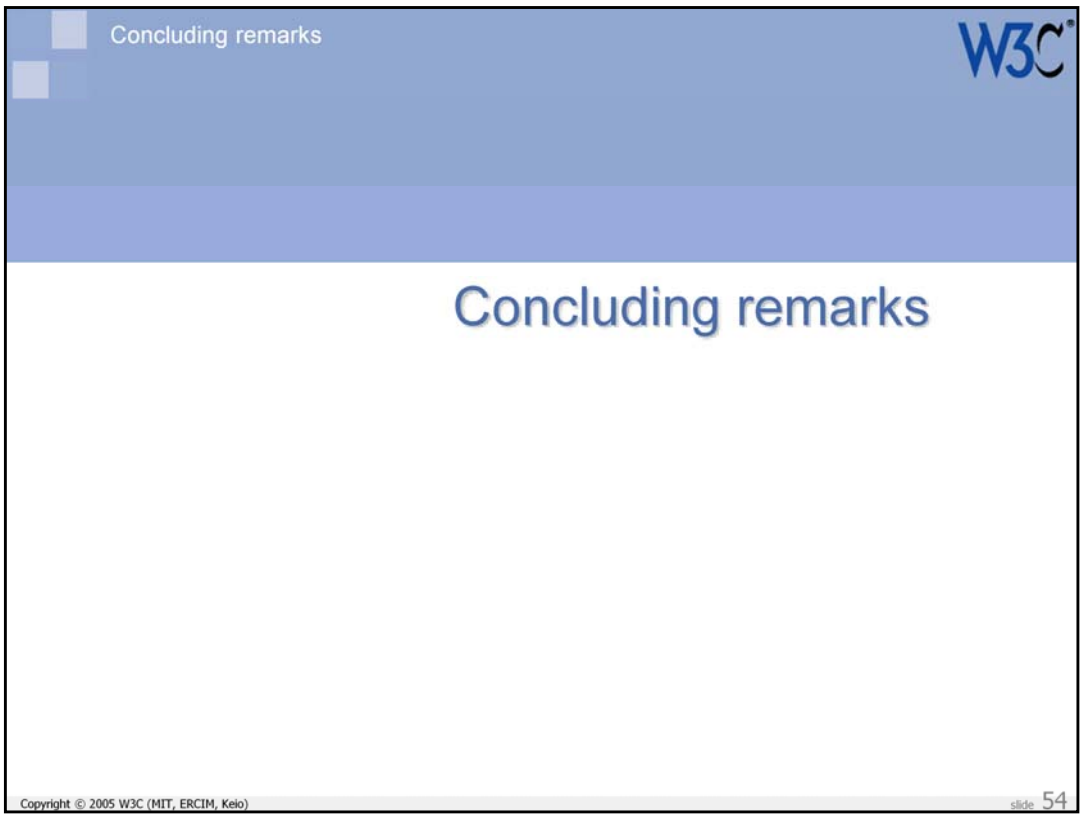


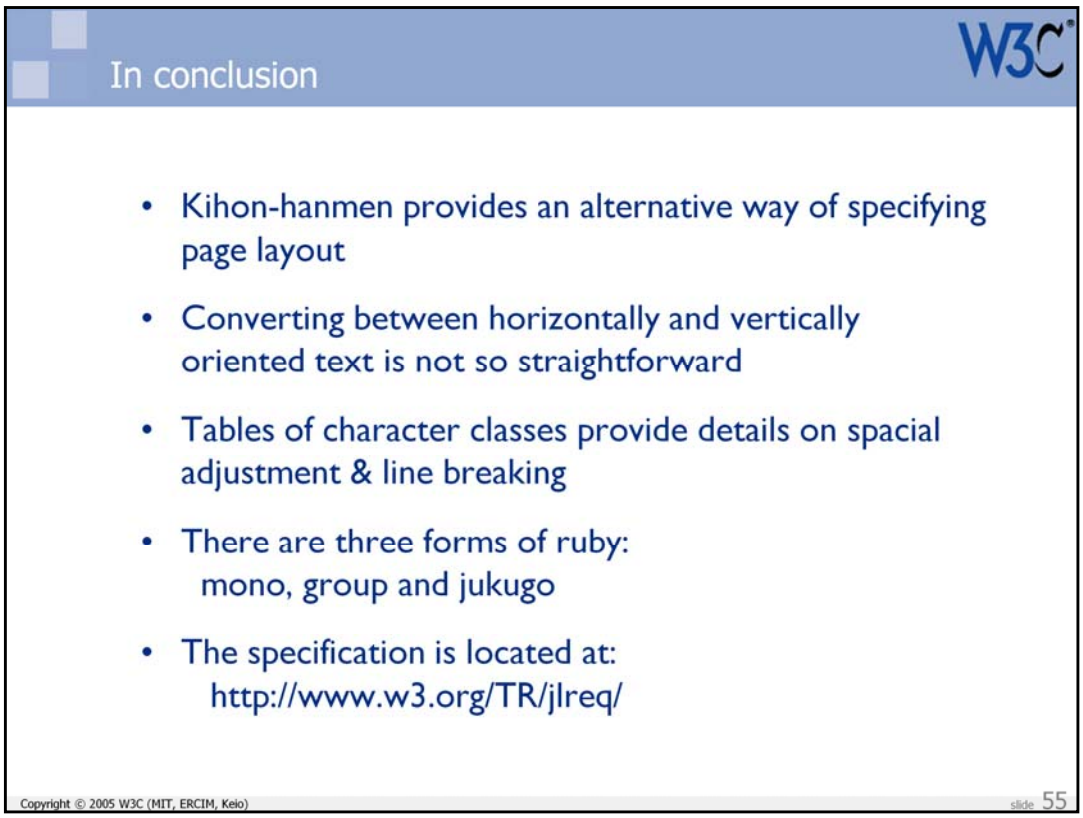
As for group or mono ruby, the overlaps are only allowed if the adjacent characters are not kanji; that is the case in these examples.

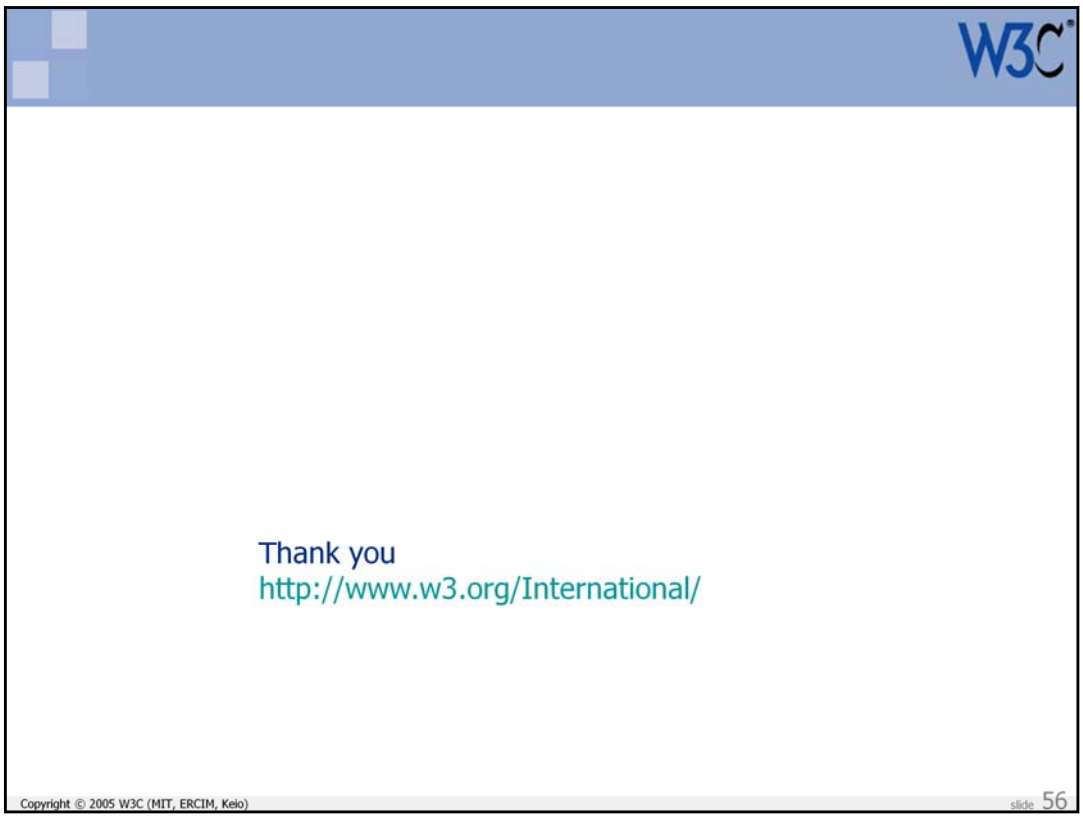


One major difference between jukugo and group ruby is that a line break may not occur within a group ruby sequence. For jukugo ruby, on the other hand, line breaks can occur and the layout reverts to the mono ruby layout before and after the line break. The example shows the behavior of our previous 1-3 and 3-1 examples when a line break occurs within the jukugo base character sequence.

note that when there are 3 or more mono ruby characters at the start or end of a line, then the base characters may be moved down to give room, within the Kihonhanmen, for the ruby characters. There are other ways to set the ruby characters in that situation, but that is beyond the scope of this talk.


Concluding remarks
W3C
Concluding remarks
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 54


In conclusion
W3C
• Kihon-hanmen provides an alternative way of specifying page layout
• Converting between horizontally and vertically oriented text is not so straightforward
• Tables of character classes provide details on spacial adjustment & line breaking
• There are three forms of ruby: mono, group and jukugo
• The specification is located at: http://www.w3.org/TR/jlreq/
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 55


Thank you
http://www.w3.org/International/
Copyright © 2005 W3C (MIT, ERCIM, Keio)
slide 56