

DPFJ EPUB 3 File Creation Guide

ver.1.1.4

10/24/2025

Contents

About This Guide	3
■Purpose.....	3
■Outline	3
■Guide Composition	4
■Supplementary Materials	4
■Reference Materials.....	5
Assumed Reading System Actions	6
■Document File Basics	6
■Package Documents/OPF Files.....	6
■EPUB Navigation Documents	7
■Style Sheet Basics	7
■Text	8
■Images	9
■Cover Images	10
■Page Media Margins.....	10
■Other HTML Elements	11
■Other CSS Interpretations.....	11
■Fixed Layouts	12
■Other	12
■HTML elements and CSS properties this guide does not assume RSs will handle.....	12
Items Publishers Should Prepare	15
■Notification of and measures for dealing with image and content document size restrictions.....	15
■Handling items designated, "follow publisher's instructions"	15
■Templates and style sheets for standard pages independently created by publishers.....	15
■Rules for simplifying appearances.....	15
Basic Creation Description Items	17
■Create data that does not cause errors in the latest epubcheck	17
■Basic Folder Compositions and File Names.....	17
■File Specifications.....	17
■Simple Coding Rules	18
List of EPUB-composed File Templates.....	20
A. Reflowable	21
B. Fixed Layouts	34

List of Forms	40
■ Formal Paragraphs.....	40
■ Line Breaks, Blank Lines.....	40
■ Horizontal text in vertically oriented pages.....	40
■ Text direction when vertically oriented.....	40
■ Rubies (Phonetic readings of kanji printed in smaller sized text next to the kanji)	41
■ Designating Special Text.....	41
■ Text Decoration.....	42
■ Emphasizing/Negating Text.....	42
■ Images	43
■ Headings	45
■ Aligning the positions of in-line elements.....	46
■ Aligning Lines.....	47
■ Lowering Text and Indenting.....	48
■ Line and Text Spacing	50
■ Separators.....	52
■ Links.....	52
■ Handling Boxes.....	54
■ Ruled Lines	56
■ Aligning Block Elements.....	57
■ Page Breaks in the Same File.....	57
■ Designating Colors	58
■ Wraparound	60
Default CSS Files.....	61
■ Style Sheet Compositions.....	61
■ CSS File Operation Rules	61
■ Creating and adding standard classes	62
■ Customizations	64
■ Creating your own CSS files	65

About This Guide

■Purpose

This guide is for members of the Digital Publishers Federation of Japan (DPFJ), formerly known as the Electronic Book Publishers Association of Japan and the Digital Comic Association, which merged in 2021 and was renamed in 2022. It outlines the process for producing EPUB 3 trade books.

There are several methods for producing EPUB 3 files, which has led to hesitation among DPFJ member companies. Furthermore, developers of Reading Systems (often referred to as viewers or hereafter as RS) are often uncertain of the publishers' intentions. This guide was created to resolve these issues by providing a clear direction for EPUB 3 production as a unified reference for all DPFJ members.

Unless otherwise specified, this guide defines EPUB 3 as EPUB 3.3, which was adopted as a W3C Recommendation (international standard) in 2023.

W3C EPUB 3.3

<https://www.w3.org/TR/epub-33/>

■Outline

This DPFJ EPUB 3 File Creation Guide primarily describes publications with simple formats that do not feature such things as mixed text directions and columns, as reflowable electronic publications. It does not address layouts that place text and images in arbitrary places, feature designs in which text wraps around images, or have fancy headings and the like.

However, it does touch upon simple fixed layouts as one method of presenting readers layouts that are still difficult to recreate as reflowable electronic publications, such as anthologies and technical documents containing sections with numerous charts. We also refrained from covering the usage of SVG, thought to be a little too daunting to use for a reading system (RS), when addressing reflowable publications in this guide. For this reason, there are a few restrictions when fitting images to pages. For example, you cannot enlarge images beyond their original sizes, finely control the size and location of images, or draw borders around images that have been fitted to page.

However, since some RSs internally switch processes with reflowable and fixed layouts, we chose a method to enlarge/reduce (fit) image files to page sizes using the SVG expand and contract function when describing fixed layouts types.

We also use the SVG method for image maps (clickable maps), which tend to cause problems in fixed layouts. Currently, coordinates in reduced display images are handled differently depending on the drawing engine in the HTML map element method. This is one of the reasons why we use SVG in fixed layouts.

We expect these issues to eventually be resolved. For the purposes of this guide, we only briefly address image placement. However, as it stands now even the interpretation of, and degree in which, these things are handled differ based on the RS, and in some cases images are not displayed in the same way. This is not only an image issue; the same problem exists when designating text and lines as well.

Despite the fact that EPUB employs standard Web technology, Web browser interpretations of basic elements continue to differ from vendor to vendor, just as they did nearly a decade ago.

Moreover, unlike Web browsers, the specs of some electronic publication RSs are not open to the outside, and data cannot always be easily accessed to verify actions, so it is difficult to describe data that expect those common actions.

For this reason, this guide assumes simple RS interpretations of very basic designations used to recreate typesetting expressions in its descriptions, while referring to the actions of Radium, provided by Radium Foundation. In consideration of the performance of commercial RSs and interpretation commonalities, this guide supposes the use of only a few functions out of all the functions EPUB 3 is capable of employing.

We have also prepared a style sheet set for your use, which the content of this guide more or less adheres to. In a time when functions usable between multiple RSs lack stability, this approach primarily serves to prevent the unknowing inclusion of properties and usages of CSS that are not recommended or assumed in this guide in publication data designations, and to prepare logical directional class names that enable vertically and horizontally oriented pages to

use the same specified format. We expressed document compositions with as little HTML as possible, and did not take the approach of using class names, ids and the like to finely regulate the compositions of works.

It is not good when display devices do not allow authors and editors to always show what they want to display. In order to make publications more convenient for readers by reflowing text and allowing text size to be dynamically changed, layouts must suffer to a certain extent in terms of restrictions on the properties of electronic publications and their current functions. However, this does not mean that the center location of text and images should differ based on the RS, the direction of certain sections of text should differ from other sections, and display control methods should vary from RS to RS. These types of inconsistencies help no one, and should be resolved as soon as possible.

We expect that authoring tools, which are growing in importance and should aid the creation of publication data, will be able to stably output source that displays data the same way on any RS without the need for special measures or complicated descriptions.

It is our hope that this guide, while not enough to satisfactorily digitize commercial publications including their functional aspects, helps to reduce turmoil such as that described above.

■Guide Composition

Assumed Reading System Actions

Before listing publication data, this guide first lists assumed RS behaviors in this section. At present there are inconsistencies between commercial RSs. This section primarily describes inconsistencies that tend to impede the creation process, and basic actions that must be ensured to avoid problems.

Items Publishers Should Prepare

Lists items beyond the scope of this guide that each publisher should prepare as needed.

Basic Creation Description Items

Lists fundamental rules to follow when describing publication data.

List of EPUB-composed File Templates

A list of basic page templates that can be used to describe publications.

Also includes a sample method to recreate publications on RSs that mix vertical and horizontal text in order to display horizontally centered vertical text--a format many publishers request, but one which this guide does not recommend--and its restrictions and limits.

Format List

A list of formats for decorating text, inserting images, and performing similar actions within pages used in the previous section's templates, along with brief descriptions of each format.

This guide does not use the half-width blank space before the end slash in the XHTML empty element, which is used for compatibility with old HTML browsers that do not correctly interpret XHTML in the templates and formats listed.

Default CSS Files

A description of the style sheet set prepared by default to use with the formats in the previous section, along with their operating rules and methods to perform simple customizations and the like.

■Supplementary Materials

- book-template_YYYY-MM-DD.epub & fixedlayout-template_YYYY-MM-DD.epub

These actual EPUB files contain the templates mentioned in the section of this guide entitled, "List of EPUB-composed File Templates". Feel free to alter and use them freely according to your needs.

Note that they do not contain horizontally centered vertical text, which this guide does not recommend.

Also note that pages whose details and appearances differ from publisher to publisher, such as title pages and imprints, are designated only with the words, "Title Page" or "Imprint".

- dpbj-sample_YYYY-MM-DD.epub

A sample work created following this guide. It contains rubies (small, annotative glosses), decorations, location changes, annotations and other designations regardless of their relation to actual work content, in order to test the functions. It does not contain all available functions, but feel free to use it as a reference for actual publications, or to check RS actions.

- CSS-Function-List_YYYY-MM-DD.pdf

A list of classes included in this guide's default style sheet. It lists brief descriptions and usage method examples of each function. Not all functions that list size values were included, but rather only their patterns. Refer to the "Format List" section of this guide for more detailed information.

■Reference Materials

- Reference-Information_YYYY-MM-DD.pdf

We have referred to particular points of view of the new EPUB specification, though this guide does not assume using practically. Remember that new elements adopted by EPUB are not yet taken in by RSs; understand well that the actual behavior is yet unknown, and then refer the examples for creation hereafter.

- Items-RSs-are-expected-to-have-in-the-future_YYYY-MM-DD.pdf

Though this guide does not assume using practically, we have noted the difficulties on creation, and the points to correspond promptly when having additional RS functions, EPUB specification renewal, etc.

Assumed Reading System Actions

This guide assumes the reading system (RS) is capable of performing the actions listed in this section. The following meta data is listed on package documents (OPF files) as a keyword to notify the RS that the electronic publication data conforms with this guide.

```
<meta property="ebpaj:guide-version">1.1.4</meta>
<meta property="dpfj:guide-version">1.1.4</meta>
* "ebpaj: http://www.ebpaj.jp/" and "dpfj: https://www.dpfj.or.jp/" are listed in the package's prefix attribute to declare the prefix.
```

Unless otherwise specified, publication resources (XHTML, CSS and SVG) are interpreted and displayed based on Radium, which acts on Radium Custom Chromium binary for Mac OS X (Radium-Chromium), distributed by the Radium Foundation.

When the name "Radium" appears by itself in this guide, it refers to the aforementioned Radium that acts on Radium Chromium.

For the sake of convenience, general descriptive files that comprise publication data, such as package and navigation documents, content documents, and style sheets, are all referred to hereafter as "document files."

■ Document File Basics

Text Code

The text code to be used in document files is UTF-8.

Documents will be interpreted the same way regardless of the presence of BOM.

*This guide recommends saving document files without BOM.

Line Break Codes

"CR+LF," "CR," and "LF" should all be correctly interpreted as line break codes in document files.

However, this does not apply to files in which different line break codes have been mixed together.

*This guide recommends you avoid mixing different line break codes within the same file.

Handling spaces, line breaks, comments, etc. in the source

In general, the XHTML descriptive rules and their interpretations should be followed for document files.

Ignore comments in the source, allow optional ordering of attributes within elements, include one or more space/line break/tab between attributes, text should be appropriately processed; maintain the same level of strictness and flexibility as modern browsers which supports Web Standards.

container.xml in META-INF

This guide does not list an example of two or more OPF files. However, the EPUB 3 specifications should be followed for two or more OPF files as well, to appropriately process them.

■ Package Documents/OPF Files

Preserving the Page Progression Direction

The page progression direction of publication data should follow the "page-progression-direction" listed in the package document's spine element, regardless of the "-epub-writing-mode" designation listed in the content document of the style sheet.

The page progression direction of XHTML file content should follow the "-epub-writing-mode" designated in each XHTML file's body element.

The "-epub-writing-mode" designated in the html element should appropriately follow the body element.

For example, if the "page-progression-direction" is "rtl (right to left)," and the XHTML's "-epub-writing-mode" is "horizontal-tb," text will appear from "left to right" in the screen, but new pages should increase from "right to left."

Preserving the spine element's designation

The order should correctly reflect the order designated in the spine element.

The "yes" "no" designation in the itemref element's "linear" attribute should be reflected. If the "linear" attribute is

designated as "yes," it should not be hidden even if it is the cover page.

The "page-spread-right" "page-spread-left" designation in the itemref element's "properties" attribute should be reflected.

Publication data that does not designate anything in its spine element should not be shown as a book page.

Functions do not rely on the use of elements that have been scrapped

The guide element was discontinued, so this guide does not describe how to handle functions that rely on it.

Handling Metadata

If an RS features a function that shows the <dc:title> information, we assume all the listed information will be shown on the screen.

If an RS has a function that shows <dc:creator> information, and there are multiple <dc:creator>s, this guide assumes all the listed content will be displayed on the screen somewhere in the RS. (We leave things like the codes of linking together multiple author names and the displays of role notations entirely up to the RS.)

Follow the publisher's instructions when deciding whether to separate each of the multiple authors' names, or list them all in a single <dc:creator>. If separating them, the publisher shall provide the role values of the authors and designate the order in which they appear.

There is no established code system used for file IDs ("unique-identifiers"). (Follow the publisher's instructions. If no instructions are provided, insert uuid.)

If no revision date is specified, use the scheduled delivery date so that later files can be easily managed.

It is best to hide revision dates from readers.

■EPUB Navigation Documents

Preferential Interpretation of Navigation Documents

If a navigation document and a ncx file are bundled together, interpret the navigation document is given priority.

This guide does not describe how to handle functions that rely on obsolete ncx files.

Display Navigation documents

The RS shall handle how to display navigation documents.

These specifications do not assume whether or not items other than links are included in navigation documents.

■Style Sheet Basics

CSS Property Restrictions and Standard of Interpretation

In light of each company's RS support for CSS, we narrowed down the amount of CSS properties expected to be used when creating publications to the minimum. Regarding each of the elements, which are interpreted differently by RSs even though they are basic properties used to display books in Japanese, and which are causing problems when creating publication data that can be displayed with the same source, we primarily use Radium's interpretation as the description standard, and refer to existing RS interpretations to a certain extent for areas not covered by Radium. When using properties not covered in this guide, each publisher shall make adjustments with things like the RS's display and other designations within the guide.

Using the Default Style Set

A basic style set has been prepared for this guide, which mainly describes the classes used in the set. However, since this guide permits the customization of CSS files, some files may add new classes, change the names of existing classes, include existing classes with other properties, etc. RSs shall not assume class names are fixed and rely on them as such, but shall appropriately support the designated CSS properties.

To learn more about the types of properties that may be designated by publication data in accordance with this guide, refer to the below section entitled, "HTML Elements and CSS Properties that Assume RS Support."

Alternative Style Sheets Not Used

The RS shall address the process of switching between vertically and horizontally oriented pages.

If the RS prepares this type of function, we expect it to observe the following two points:

- Show the orientation designated in the work's data as the default.
- When changing the orientation to a different direction than that designated in the work's data, display a message in an appropriate location such as the text introducing the RS's features or the help section stating that changing the orientation may contradict the author or the producer's intentions.

We do not support the process of allowing users to change the orientation direction for the following reasons: at present,

there is a shortage of environments that actually display altered orientations; there is nothing to act as a producer/supervisor's index; and a CSS property to enable the use of logical orientation designations to assist in designs that do not rely on an orientation direction is not yet available. This makes it extremely difficult to allow readers to freely change the orientation, at times requiring as much effort as it takes to design a new publication.

CSS Property's Preferential Interpretation of the "-epub-" Prefix

A CSS property, first adopted in the CSS3 before it was recommended, prioritizes files with "-epub-" prefixes in its interpretations. However, in consideration of current RS specs and simple Web browser checks, properties requiring the "-epub-" prefix also include the "-webkit" prefix.

*Continuing to add prefixes for each vendor, whom we expect to increase in number, will prove to be a difficult task. In the future, when the WebKit's CSS description method and interpretations change, RSs that prioritize "-webkit-" in their interpretations may encounter difficulties. It would be best if RSs prioritized files with "-epub-" prefixes regardless of the order they are listed in CSS files.

Only properties that are mentioned in the EPUB 3.0.1 specification can use "-epub-" prefixes.

Be aware that there is no meaning in just rewriting properties that can use "-webkit-" prefixes by "-epub-" prefixes in browsers, or anything else.

Adopting @import Rules

Basic XHTML templates only read main CSS in link elements, using @import rules and reading each necessary CSS file from inside the main CSS file in order to simplify and unify the CSS file designated descriptions on the XHTML side, and to secure flexible CSS customization. However, this does not apply when creators want to change CSS files to read depending on the page (such as when they want to read style sets by publisher only on certain pages.). RSs should also be able to call multiple CSS files from XHTML.

We do not recommend additionally using @import in files that were read with @import.

Appropriately ignore designations of unsupported fonts

Fonts not supported by RSs should be appropriately ignored when designating fonts.

Likewise, font set fonts unsupported by RSs that are designated by @font-face should likewise be ignored.

Designations for html elements

As a general rule, only orientation direction and font designations are performed in html elements.

Both are appropriately passed along to body elements.

RS Overwriting of the Default Style Sheet

The RS's set default style sheet can be overwritten on the publication data side.

It would be best if the default style sheet information is disclosed to the publisher.

■ Text

Character Set

At minimum, the following character set will be supported:

JIS X 0213:2004 (Surrogate Pair area of Unicode are also included)

Fonts

At minimum, the following two fonts will be usable.

Monospaced Ming-cho-type font

Monospaced Gothic-type font

Furthermore, the above fonts will be assigned to generic font families as follows:

Serif: Ming-cho

Sans-serif: Gothic

When the font size is the same, the em size of both fonts will be equal.

Their em sizes will be equal regardless of differences in full-width and half-width characters.

The em sizes will be fixed even if -epub-text-orientation is used in the style sheet to change the text direction.

The display width of each font's characters will not be markedly different.

*It would be best if fonts were monospaced rather than proportional when reproducing basic typesetting expressions in vertical orientations such as positional adjustments for indentations, etc. However, be aware that positional adjustments such as shrinking text when horizontally oriented may not produce the desired effect.

Text Direction when vertically oriented

We will assume the display to be based on the document below, presented by Unicode Consortium.

"Unicode® Standard Annex #50 Unicode Vertical Text Layout"

<https://www.unicode.org/reports/tr50/>

The text direction in vertical orientations is designated as an item changed in the below CSS properties. Text whose direction has been altered shall conform to the above document.

Upright: `-epub-text-orientation: upright;`

Rotate 90° right (sideways): `-epub-text-orientation: sideways;`

When the value "rotate-right", which was in the EPUB 3.0 specification, is used to designate rotation, or if also "sideways" is used, it should make the same move as sideways.

*Presently, several RSs do not center half-width text set to the upright orientation in the same location. Due to this, this guide was forced to use `-epub-text-combine` instead in some areas.

As its name suggests, `-epub-text-combine` is a property that combines text. If possible, it should not be used to change text direction.

Also, in the CSS upright designation, a character would be displayed by its vertical glyph if it has one regardless of the established directions such as "Tr" "Tu" "R" "U", so be aware of the characters whose shapes may be changed due to the text orientation.

Vertical-align in vertically oriented text

We assume images and upright text would be displayed (centered) by "The central baseline" from "4.2 Text Baselines" of "CSS Writing Modes Level 3 W3C Recommendation, 10 December 2019".

<https://www.w3.org/TR/css-writing-modes-3/#central-baseline>

*As with small script text, there are times when you may want to line up text that has been reduced in size on either the left or right side of vertically oriented lines. To that end, it would be helpful if the "text-top" and "text-bottom" designations were used to line up text on the right & left sides (when vertically oriented) of the parent element's text inline box, rather than govern things like line height (it would be preferable if the position of designated text did not shift even if text in the same line is enlarged and expands the expansion inline box). This guide recommends using "super" and "sub" just to be safe, but display results may differ slightly from expectations since they both protrude a little from lines.

Automatic Expansion of Line Spaces

As a general rule, lines are evenly spaced so long as the publication data designation is not changed. However, if the last line does not fit within the page, there is no choice but to allow it to be sent to the next page leaving blank space afterwards.

Normally when data is created, allowances are made for the effects of size and width changes in the event the font size of some text in a line is enlarged, or text flows over a line, such as when inserting horizontal text in a vertical line. However, end-of-line return positions cannot be foreseen in reflowable publications, making them difficult to manually avoid. For that reason, it is preferable if spaces are maintained as much as possible. However, if text overlaps with text or things like rubies or emphatic dots in neighboring lines, then the spaces between lines should automatically expand.

Line Boundary Character Check

In light of the current state of RSs produced by various manufacturers, this guide assumes restricted processes will depend on the RS.

■ Images

Image Types

JPEG, PNG, GIF, and WebP images shall be usable.

It would be preferable if PNG and GIF images can use transparent backgrounds.

* Note: The WebP format is available for use starting with EPUB 3.3, and support from RS is expected. However, before using it, be sure to confirm that your distribution platform or RS supports it.

Images of Non-standard Kanji Characters

These images shall be displayable in the size of one character of text, using this guide's image reduction setting.

There are no restrictions on image sizes.

Recommended Value

Image Size: 128px × 128 px

Image Format: 8bit transparent background PNG

Anti-aliasing: None

*We recommend an 8bit PNG saved with its background color made transparent, in consideration for RSs that allow users to freely change the background color. However, if applying background color to the text, it is probably safer to use non-standard kanji images with the same background color since some RSs may not support transparent background images.

Maximum size value designation when setting image and block element sizes

Max-height and max-width actions are two actions that differ in interpretation between current browsers. RSs shall be based on Radium for publication data that follows this guide.

These are mainly used to fit images to pages, and to insert non-standard kanji images.

This guide presupposes maximum size designations based on Radium actions, until designations appropriate to page fittings with CSS 3 and later versions are adopted and used in EPUB specifications.

Determining whether or not a replacement element such as an image, or a block/inline block element, crosses over a page

An "element's display size," described below, is defined as its designated size in a style sheet or similar document, if one exists. As stated earlier, a size interpretation based on Radium will be conducted in order to designate the { max-width: 100%; } size, designated in an element.

If an element in a text line is larger than the text width, and if the element to be displayed does not fit into "the space allotted for the element to appear," the entire line in question shall be sent to the next page.

If an element that should be shown does not fit into "the space allotted for the element to appear" in an RS page, the determination of whether or not it crosses over to another page will depend on whether the page progression direction width (horizontal width for vertically oriented pages) is larger or smaller than the width of the RS's page display field's page progression direction width, regardless of whether it is inline or block.

If equal to or smaller than the page progression direction width Display elements that do not fit in the allotted space on the next page so they do not cross over to another page.

*If an image has a page fit designation, fit it to the page and display it after being sent to the next page.

So long as an element does not have a page fit designation on the side that isn't on the page progression direction side (its height, for vertical orientations), no measures can be taken if the element is larger than the remaining screen space and runs off the screen in the direction of the last line without being displayed.

If the element is larger than the page progression direction's width, display it crossing over to the next page.

User enlargement of reduced images

Users shall be able to use controls such as pinch in to enlarge images that have been reduced in size, due to size designations and page fitting designations, back to their original sizes.

*This is because we want to allow them to read things like text in images even on small screens.

■Cover Images

Preparing alternate images in bookshelves, etc.

Since not all publications have a cover image (for rights clearance reasons, etc.), it would be preferable if RSs prepared an alternate image to show in bookcases for publications without cover images, or took similar measures to enable the smooth execution of operations.

File Names

If publishers do not specify a file name for cover images, RSs shall label them all identically (cover.jpg) to speed up the display of thumbnails.

■Page Media Margins

Body element margin designations

Currently, most RSs insert margins that cannot be controlled from the publication data side. For this reason, the default for the margin and the padding in the body element are 0.

Arbitrary additions/deletions of margins in body display field by RSs

RSs shall not independently add margins that affect the usable screen size in the body element. Similarly, RSs shall not arbitrarily push together designated margins, paddings, and blank lines in publication data.

*For example, actions such as automatically adding margins that cannot be deleted only at the beginning of files, or arbitrarily pushing together designated margins, may impede authors and creators from conveying their intentions to readers, and should be avoided at all costs.

■Other HTML Elements

 for blank lines

Lines with only
 will be treated as a single blank line to facilitate things like blank line placement, the reuse of data for printing, and handling in Japanese publications.

Also, codes using p and div elements with margin and padding set to zero, such as <p>
</p> and <div>
</div>, will likewise be treated as a single blank line.

Rubies (printed readings of kanji (Chinese characters))

The designation of "<ruby> 漢<rt> か ん (KAN)</rt> 字<rt> じ (JI)</rt></ruby>" (kanji combination ruby-like designations) shall be usable.

When rubies and circled characters/characters marked with emphatic dots are simultaneously designated, rubies will be given priority.

Non-standard kanji images shall also allow ruby designations.

The following expressions can be inserted in ruby strings shown as "<rt>ruby string</rt>".

- All normal text (including western text and numbers)
 - Number/text references
 - Non-standard kanji images
 - Text orientation designations (-epub-text-orientation)
 - Horizontal text in a vertical line (-epub-text-combine and -epub-text-combine-horizontal)
- *In order to use text direction designations as well

We did not consider decorative aspects such as colors and sizes in ruby strings this time.

We did not consider vertical-align changes (superscript, subscript, etc.) this time, either.

Links in Pages (Anchor Links)

When clicking, tapping, etc. on a link designated by "text" (if the file name is in the same file it can be omitted), the page will jump to the element designated by "<element name id='\"anchor name\"'>". Do not use full-width characters or spaces when actually using file and anchor names.

Whether or not the elements in the jump destination to are always displayed on top, or a page including those elements is simply displayed without changing the pre-calculated page's arrangement is up to the RS.

*The former was more natural in scroll media, but the latter may be better at times in page media.

Nav & List Elements

This guide does not assume the use of nav elements outside navigation documents, and list elements such as ol and li. As described in EPUB Content Documents, list numbers are not shown for list elements in navigation documents.

■Other CSS Interpretations

Emphatic Lines

Considering the usage of "-epub-text-underline-position" from EPUB 3.0.1, designate

{ -epub-text-underline-position: under left; } in the body, and { text-decoration: underline; } to the line on the right side of the text and { text-decoration: overline; } to the left when in vertical orientation.

Since the reflection of "-epub-text-underline-position" is not guaranteed at the present moment, do not designate { text-decoration: underline; -epub-text-underline-position: right; } to the line on the left side of the text.

{ -epub-text-underline-position: auto; } is not assumed to be used at the present moment.

text-decoration lines will not be drawn until rubies and dots to emphasize words.

When using side lines together with horizontal text included in vertical orientations, steps shall be taken to ensure the lines do not appear in awkward locations.

text-decoration is used this time assuming all these issues will be resolved.

In CSS 2.1, text-decoration does not work in the inline-block element, so emphatic dots should not be drawn in non-standard kanji images and annotation symbols designated by inline-block in the CSS file.

Horizontal Text in Vertical Orientations

{ -epub-text-combine: horizontal; } and { -epub-text-combine-horizontal: all; } are assumed to designate up to 3 half-width digits to become horizontal text in vertical orientations.

Horizontal text strings in vertical lines are treated as a single character.

Ex) When a side line (text-decoration) is designated as horizontal text in a vertical line, only one line appears on the outer side of the text when vertically oriented.

When an emphatic dot is designated as horizontal text, the horizontally oriented text string is treated as a single character, and a single dot is shown next to the entire character string.

When a ruby is designated as horizontal text, the horizontally oriented text string is treated as a single character, and is shown as stipulated by the rules governing rubies.

In the specification of EPUB 3, "-epub-text-combine-horizontal" is adopted due to the change of property name in [CSS3WritingModes-20121115],

in [CSS3WritingModes] (20140320), the present Candidate Recommendation, but considering that the property name had been changed yet again and became "text-combine-upright", we assume "-epub-text-combine" should be used by designating horizontal text in vertical orientations for the time being.

Moreover, when "-epub-text-combine-horizontal" is used, it should be understood as designation of horizontal text in vertical orientations. Also, when using these, we recommend to designate { text-combine-upright: all; } additionally, considering the convenience of browser confirming from now on.

Hidden Designations

{ display: none; } shall be correctly interpreted.

However, at present we recommend either not including elements that should not be shown to readers in publication data, or using comment out, which is safer, for safety purposes.

We expect { visibility: hidden; } to be correctly interpreted by replacing the deleted text with blank spaces.

However, since there probably will not be any cases that require the use of visibility, this guide, which does not anticipate things like the use of Java Script and mouse hovering processes, put off including classes that enable the use of visibility in CSS files.

■Fixed Layouts

Support to Fixed layouts shall be limited to works comprised entirely of images.

Fixed layouts comprised entirely of images employ the SVG wrapping method.

SVG will be written directly in XHTML. The format of images that can be included will be the same as in reflowable publications.

Image maps (clickable maps) will also accommodate the resizing of images using SVG.

Create data so that pages oppose each other, except for the cover page.

Only cover pages will designate properties="rendition:page-spread-center" in the spine-itemref element, based on the EPUB Fixed Layout designation method, so they shall always be displayed by themselves.

Cover pages will be designated to appear spread across the center of the screen, but they do not need to be placed in the center so long as the spread location is not later distorted.

■Other

Scripting

This guide does not assume the use of JavaScript in light of the current state of the various RSs available.

■HTML elements and CSS properties this guide does not assume RSs will handle

【HTML】

Document metadata

base element

Scripting

script element / noscript element

Sections

section element / article element / aside element / header element / footer element / address element

Content Grouping

blockquote element / ul element / dl element / dt element / dd element / figure element / figcaption element

Attaching meaning on the text level

em element / strong element / pre element / sub and sup element / i element / b element / u element / s element / small element

cite element / q element / dfn element / abbr element / time element / code element / var element / samp element / kbd element

mark element / rp element / bdi element / bdo element / wbr element / rb element / rtc element

Revisions

ins element / del element

Embedded content

map element / area element / iframe element / embed element / object element / param element / video element / audio element

source element / track element / media element / canvas element / MathML / SVG (other than fixed layout usage)

Table data

table element / caption element / colgroup element / col element / tbody element / thead element / tfoot element / tr element

td element / th element

Forms

form element / fieldset element / legend element / label element / input element / button element / select element

datalist element / optgroup element / option element / textarea element / keygen element / output element

progress element / meter element

Interactive elements

details element / summary element / command element / menu element

【CSS】

Values

ex / in / cm / mm / pt / pc

Selector

E:focus / E:lang(c) / E:first-child / E:first-line / E::first-line

E:first-letter / E::first-letter / E:before / E::before / E:after / E::after

E[foo^="bar"] / E[foo\$="bar"] / E[foo*="bar"] / E:root / E:nth-child(n)

E:nth-last-child(n) / E:nth-of-type(n) / E:nth-last-of-type(n)

E:last-child / E:first-of-type / E:last-of-type / E:only-child / E:only-of-type

E:empty / E:target / E:enabled / E:disabled / E:checked / E:not(s) / E ~ F

@Rule

@page / @page:left / @page:right / @page:first

Color/background

background-attachment / background-image / background-position / background-repeat

Font

font-size-adjust / font-stretch / font-variant

Text

text-shadow / text-transform / white-space / word-spacing

Width/height

min-width / min-height

Display

direction / visibility / clip / overflow / unicode-bidi / z-index

display: list-item; / display: table; / display: inline-table / display: table-row-group

display: table-header-group / display: table-footer-group / display: table-row

display: table-column-group / display: table-column / display: table-cell

display: table-caption

Page media

page / size / marks / orphans / widows

List

list-style / list-style-type / list-style-position / list-style-image / marker-offset

Wraparound
 float / clear
 Position
 position / top / right / bottom / left
 Table
 border-collapse / border-spacing / caption-side / empty-cells / table-layout
 Insert
 content / quotes / counter-reset / counter-increment
 Outline
 outline / outline-color / outline-style / outline-width
 CSS 3.0 Speech
 -epub-cue / -epub-pause / -epub-rest / -epub-speak / -epub-speak-as / -epub-voice-family
 CSS Text Level 3
 -epub-hyphens / text-transform: -epub-fullwidth / text-transform: -epub-fullsize-kana
 CSS Writing Modes Module Level 3
 caption-side: before / caption-side: after
 CSS3 Multi Column
 column-width / column-count / columns / column-gap / column-rule-color
 CSS 2.0
 list-style-type: cjk-ideographic / list-style-type: hebrew / list-style-type: hiragana
 list-style-type: hiragana-iroha / list-style-type: katakana / list-style-type: katakana-iroha
 EPUB 3
 -epub-ruby-position / display: oeb-page-head / display: oeb-page-foot

Items Publishers Should Prepare

This guide only serves as a guideline for describing simple publications.

Creators are responsible for preparing materials necessary for creating publications, such as original drafts and logo data, as well as establishing detailed creation rules for each publisher such as delivery data compositions and file names. This section lists examples of some things you may need to prepare in advance when describing publication data content.

■ Notification of and measures for dealing with image and content document size restrictions

There may be limits to the file sizes of images and content documents such as XHTML due to performance restrictions, depending on the device. If necessary, list size limits for each file and provide instructions on what to do if those limits are exceeded.

■ Handling items designated, "follow publisher's instructions"

If necessary, provide individual instructions for optional items such as those labeled, "follow publisher's instructions" in this guide.

Ex) "Position all cover pages on the left side when displaying pages spread open."

■ Templates and style sheets for standard pages independently created by publishers

When preparing templates common with company products, be sure to use HTML elements and CSS properties assumed by this guide to create them.

If using templates and style sheets by publisher, carefully consider each of their display contents.

■ Rules for simplifying appearances

Each publisher should prepare rules for transferring information such as complex appearances from paper books.

[Elements that often cause trouble]

- Image insertion position
- Simplification of appearances such as heading sizes and font replacements
- Heading level setting rules (<h1> is the cover heading, etc.)
- How to process covers, headings, contents, etc. that have fancy designs (convert them to text only, or images, etc.)
- Whether to leave in or delete page numbers on table of contents pages and annotation pages.
- Letter shapes (especially personal pronouns such as the names of people, etc.)
- How to process spread-out displays
- How to process things like dual pages, foldout pin-ups, etc. (change the direction, etc.)
- Processing wide pages and narrow pages
- Top- and bottom-aligned text in the same line
- Designations when vertically oriented text cannot be horizontally centered
- Whether to utilize blank pages or remove them
- Designations such as removing throwaway lines used to send headings to the next page, insert page breaks, etc.
- Whether to utilize difficult settings such as centering two lines, or to simplify them
- Whether to scan and utilize section divider images, or to replace them with codes, etc.
- How to process inserted notes
- What to do with offset-like processes
- How much to utilize ruffled borders
- How to process columns
- What to do with text background images (especially images designed for fixed layouts, such as manuscripts)
- What to do with contact information, addresses and the like in imprints and postscripts
- Advertisement processing

- Treatment of things like "refer to page ●."
- What do to with indexes

In particular, other people will not know the ratio of text size to the original, so you should try to designate this individually if you want to be accurate.

If you want to preserve the size relationship even when the size ratio changes, we recommend you establish such a rule in advance.

Basic Creation Description Items

Unless instructed otherwise, follow the rules below when creating data.

Bear in mind the file/folder name and source formulation rules are merely guidelines when creating data with a given standard. We recommend creating data under consistent rules, since other people may encounter difficulty when making revisions if each creator's style varies drastically from others.

If a publisher instructs you to do it differently, such as use a certain authoring tool to automatically generate data, follow the publisher's instructions instead.

■ Create data that does not cause errors in the latest epubcheck

W3C/epubcheck · GitHub

<https://github.com/w3c/epubcheck>

■ Basic Folder Compositions and File Names

```

root folder
├─ mimetype
├─ META-INF folder
│   └─ container.xml
└─ item folder
    ├── standard.opf
    ├── navigation-documents.xhtml
    ├── image folder
    ├── style folder
    └─ xhtml folder
  
```

- Follow the publisher's instructions when setting the root folder name.
- As a general rule, file and folder names should be in lowercase (except when otherwise instructed, such as by META-INF or control codes.)
- The name of the material storage folder shall be "item", to be consistent with <item>element in the package document (an optional specification)).
- All materials shall be placed in specified folders inside the item folder. No other folders or subfolders should be created.

Image files: "image" folder

CSS files: "style" folder

xhtml files: "xhtml" folder

- Do not change the following files (use the enclosed samples attached as-is):

mimetype, directly under the root folder

container.xml, in the META-INF folder

- At the very least, we recommend splitting up XHTML document file names for all works by their shared pages and their body sections.

*Make considerations for reducing the work involved in distinguishing files when replacing things like precautions, and for administrative tasks such as avoiding changes in link-designated URLs through the addition or deletion of images, comments, etc.

■ File Specifications

- Create illustrations and images that are spread across two pages in the original text as a single image connected by left and right pages, and fit them to the page when inserting them.

- Split up files by page breaks in the original text when creating XHTML documents.

Works with no page breaks in them should be split up in chunks about 240KB (less than 256KB) in size.

(If there is a nearby heading, split the file before it. If not, split the file where there is a blank line.)

- Label all file titles by the name of the work.

Insert the title of the content contained in a file in the XHTML document's "<title>~</title>" section.

If a file contains various types of content, such as multiple chapters, decide your own labeling rules and give instructions accordingly.

Unless instructed otherwise, insert the title of the work.

Use the work title information in the package documents (OPF files), described later in this guide.

Connect main titles, subtitles, and series names with full-width spaces.

How titles listed here will be used will be up to the RS's functions and way of thinking.

They may be displayed somewhere on the screen, as in Web browsers.

Make sure there are no mistakes in the inputted information, so it is presentable to readers.

- Only insert epub:type in cover and navigation documents.

In EPUB, you can designate the epub:type attribute in order to show a page's role.

However, at present no RSs make use of this feature, and CSS settings using epub:type are not guaranteed. For this reason, for the time being we will designate the items below, which have a high chance of being used by future systems, if only to mark it.

HTML elements that epub:type can apply vary depending on their value. When using values other than the ones below, be aware that you may not be able to designate all of them as body or section.

These shall be listed in fixed page classes listed in templates, and before navigation document ids.

Cover image page <body epub:type="cover" class="p-cover">

Navigation document <nav epub:type="toc" id="toc">

Be sure there is a <nav> that designates epub:type="toc" in the navigation document, otherwise an error will occur in epubcheck.

■ Simple Coding Rules

- We recommend the UTF-8N (no BOM) text code.
- Do not combine line break codes in the same file.
- We do not recommend HTML elements and CSS properties not mentioned in this guide.
- Do not insert comments other than those instructed by the publisher.
- Use logical direction notations for class names.

Since directions change for vertical and horizontal orientations in class and CSS, primarily use the following notations in lines.

Start of line: start (vertical: top horizontal: left)

End of line: end (vertical: bottom horizontal: right)

Front of lines: before (vertical: right horizontal: top)

Back of lines: after (vertical: left horizontal: bottom)

However, since pages that only contain images are almost always horizontally oriented, you can use rop/right/bottom/left in the overall page settings.

In CSS, we considered the center of line start and line end directions as the "center" when setting class names this time. For the sake of convenience, we labeled the center of the page progression direction (front/back of lines) as "middle."

- The order in which attributes are listed in elements within the body is: "epub:type → class → id → src / href → alt."

- To avoid complications, try not to designate class in <p> as much as possible.
- Insert a line break immediately after an HTML element in XHTML documents used for the body.

Be sure to insert line break codes immediately before and after start and end tags in block level-type elements such as <div>.

However, do not insert line break codes immediately after start tags and immediately before end tags for <p> and headings <h1> - <h6>.

Ex)

```
× <h1>
  text
  </h1>
  <div><p>text</p></div>

○ <h1>text</h1>
  <div>
```

```
<p>text</p>  
</div>
```

As a general rule, do not insert line breaks for in-line type elements (, etc.)

Do not insert line break codes for <a> if <a> is not surrounded by block level-type elements (including <p>) or .

In any event, if the number of element nests increases so much you have trouble determining their corresponding relationships, consider preparing a specialized class instead of sticking with the existing one.

For sections that may undergo stylistic revisions or changes later, such as changing the font or color of all the lines of a certain character, or enclosing all text that has been lowered with a border to make it look like a letter, it may be better to define special classes rather than designate multiple classes in the same element. If necessary, use the CSS file's customization field to create new classes. Refer to the section entitled "Default CSS Files" later in this guide to learn more about customizing style sheets.

List of EPUB-composed File Templates

■ Template and File Name Rules

Unless instructed otherwise, use the below template and file name rules.

If any other template or file name rules than the ones listed here are necessary, each publisher shall be responsible for setting them appropriately.

■ Source Formulation

Follow the publisher's instructions regarding source formulations such as breaks and indents in the source, and the order of attributes within elements.

Unless otherwise instructed, formulate the source in accordance with the following templates.

A. Reflowable

Files Required for Settings

■ Colors in templates

Gray: Sections common to all works (in principle, do not change)

Blue: Sections that are common to all works, but change for each work

Red: Sections of the work that use that template, which require special attention (in principle, do not change)

Black: Non-standard parts (vary for each work, publisher)

■ mimetype

[filename: mimetype]

-----[sample code]-----
application/epub+zip

■ container.xml in META-INF

[filename: container.xml]

-----[sample code]-----
<?xml version="1.0"?>
<container
version="1.0"
xmlns="urn:oasis:names:tc:opendocument:xmlns:container"
>
<rootfiles>
<rootfile
full-path="item/standard.opf"
media-type="application/oebps-package+xml"
</rootfile>
</rootfiles>
</container>

■ Navigation Documents

[filename: navigation-documents.xhtml]

[Notes]

- Link items and list layer compositions change based on the content of the work.
- Unless otherwise instructed by the publisher, only include links to the cover page, contents page, and imprint page.
- This guide does not address whether or not items other than links can be included in navigation documents.
- The display method of navigation documents shall be left up to RSs.
- If displaying navigation documents as content pages in the body, insert style sheet designations while referring to examples in sections like Body Pages, listed later in this guide.

-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
>
<head>
<meta charset="UTF-8"/>
<title>Navigation</title>

```

</head>
<body>
<nav epub:type="toc" id="toc">
<h1>Navigation</h1>
<ol>
<li><a href="xhtml/p-cover.xhtml">Cover</a></li>
<li><a href="xhtml/p-toc.xhtml">Contents</a></li>
<li><a href="xhtml/p-colophon.xhtml">Imprint</a></li>
</ol>
</nav>
</body>
</html>

```

■ OPF Files

[filename: standard.opf]

[Notes]

- If the RS has a function to show <dc:title>, we assume all of the listed content will be displayed somewhere in the RS.

- If the RS has a function to show <dc:creator>, and there are multiple <dc:creator>s, we assume all of the listed content will be displayed somewhere in the RS.

(The RS will be responsible for codes linking multiple authors, displaying roles, etc.)

- Follow the publisher's instructions when deciding whether to separate each of the authors, or list them all in a single <dc:creator>.

If separating them, the publisher must provide the "role" value of each author.

- We assume that kana used to arrange names in alphabetical order that are designated with "file-as" will not be shown to readers.

- No code system has been established for file ids ("unique-identifier").

(Follow the publisher's instructions. If none are given, insert uuid.)

- If no revision date is given, use the delivery date to make files easier to manage later.

- It would be preferable if revisions dates were hidden from readers.

- If no file name is given for the cover file, label all cover files the same (cover.jpg) to speed up thumbnail displays on RSs.

- For horizontally oriented works, change the <spine>'s "page-progression-direction" from "rtl" to "ltr."

-----[sample code]-----

```

<?xml version="1.0" encoding="UTF-8"?>
<package
xmlns=http://www.idpf.org/2007/opf
version="3.0"
xml:lang="ja"
unique-identifier="unique-id"
prefix="ebpaj: http://www.ebpaj.jp/"
dpfj: https://www.dpfj.or.jp/"
>
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
<!-- work title -->
<dc:title id="title">work title 1</dc:title>
<!-- author name -->
<dc:creator id="creator01">author name 1</dc:creator>
<meta refines="#creator01" property="role" scheme="marc:relators">aut</meta>
<dc:creator id="creator02">author name 2</dc:creator>
<meta refines="#creator02" property="role" scheme="marc:relators">aut</meta>
<!-- publisher name -->
<dc:publisher id="publisher">publisher name</dc:publisher>
<!-- language -->
<dc:language>ja</dc:language>
<!-- file id -->
<dc:identifier id="unique-id">urn:uuid:c1215609-6ed3-45be-b0a0-b337d4594597</dc:identifier>
<!-- Revision Date -->
<meta property="dcterms:modified">2025-06-01T00:00:00Z</meta>

```

```

<!-- etc. -->
<meta property="ebpaj:guide-version">1.1.4</meta>
<meta property="dpfj:guide-version">1.1.4</meta>
</metadata>
<manifest>
<!-- navigation -->
<item media-type="application/xhtml+xml" id="toc" href="navigation-documents.xhtml"
properties="nav"/>
<!-- style -->
<item media-type="text/css" id="book-style" href="style/book-style.css"/>
<item media-type="text/css" id="style-reset" href="style/style-reset.css"/>
<item media-type="text/css" id="style-standard" href="style/style-standard.css"/>
<item media-type="text/css" id="style-advance" href="style/style-advance.css"/>
<item media-type="text/css" id="style-check" href="style/style-check.css"/>
<!-- image -->
<item media-type="image/jpeg" id="cover" href="image/cover.jpg" properties="cover-image"/>
<item media-type="image/png" id="logo-bunko" href="image/logo-bunko.png"/>
<item media-type="image/jpeg" id="kuchie-001" href="image/kuchie-001.jpg"/>
<item media-type="image/jpeg" id="img-001" href="image/img-001.jpg"/>
<item media-type="image/jpeg" id="ad-001" href="image/ad-001.jpg"/>
<!-- xhtml -->
<item media-type="application/xhtml+xml" id="p-cover" href="xhtml/p-cover.xhtml"/>
<item media-type="application/xhtml+xml" id="p-fmatter-001" href="xhtml/p-fmatter-001.xhtml"/>
<item media-type="application/xhtml+xml" id="p-titlepage" href="xhtml/p-titlepage.xhtml"/>
<item media-type="application/xhtml+xml" id="p-caution" href="xhtml/p-caution.xhtml"/>
<item media-type="application/xhtml+xml" id="p-toc" href="xhtml/p-toc.xhtml"/>
<item media-type="application/xhtml+xml" id="p-001" href="xhtml/p-001.xhtml"/>
<item media-type="application/xhtml+xml" id="p-002" href="xhtml/p-002.xhtml"/>
<item media-type="application/xhtml+xml" id="p-003" href="xhtml/p-003.xhtml"/>
<item media-type="application/xhtml+xml" id="p-004" href="xhtml/p-004.xhtml"/>
<item media-type="application/xhtml+xml" id="p-005" href="xhtml/p-005.xhtml"/>
<item media-type="application/xhtml+xml" id="p-colophon" href="xhtml/p-colophon.xhtml"/>
<item media-type="application/xhtml+xml" id="p-ad-001" href="xhtml/p-ad-001.xhtml"/>
</manifest>
<spine page-progression-direction="rtl">
<itemref linear="yes" idref="p-cover" properties="page-spread-left"/>
<itemref linear="yes" idref="p-fmatter-001" properties="page-spread-left"/>
<itemref linear="yes" idref="p-titlepage" properties="page-spread-left"/>
<itemref linear="yes" idref="p-caution" properties="page-spread-left"/>
<itemref linear="yes" idref="p-toc" properties="page-spread-left"/>
<itemref linear="yes" idref="p-001" properties="page-spread-left"/>
<itemref linear="yes" idref="p-002" properties="page-spread-left"/>
<itemref linear="yes" idref="p-003" properties="page-spread-left"/>
<itemref linear="yes" idref="p-004"/>
<itemref linear="yes" idref="p-005"/>
<itemref linear="yes" idref="p-colophon" properties="page-spread-left"/>
<itemref linear="yes" idref="p-ad-001"/>
</spine>
</package>

```

XHTML Document Files

■ Colors in templates

Gray: Sections common to all pages and works (in principle, do not change)

Blue: Sections that are common to all pages, but change for each work and each page

Red: Sections of the page that use that template, which require special attention (in principle, do not change)

Black: Non-standard parts (vary for each page, work, and publisher)

Green "□" : Represents a full-width space

■ Text orientation

The following text orientations are designated in each page's <html>

class="hltr": horizontal orientation h (Horizontal), progression direction ltr (Left To Right)

class="vrtl": vertical orientation v (Vertical), progression direction rtl (Right To Left)

*At present, we do not assume support for "vertically oriented ltr."

*We use the horizontal orientation for pages that only contain images, in order to center the images horizontally.

■ Cover Page

[filename: p-cover.xhtml]

[Notes]

- The default is horizontally centered text beginning at the top of the page.
- Do not describe any designations other than the image display.
- It would be preferable to make cover pages transmittable by having

RSs prepare an alternate image for cover pages that do not contain cover images, since not all cover pages have images (due to rights clearance reasons, etc.).

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops"
  xml:lang="ja"
  class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body epub:type="cover" class="p-cover">
<div class="main">

<p></p>

</div>
</body>
</html>
-----
```


■ Front Matter (in this sample, a frontispiece)

[filename: p-fmatter-***.xhtml] *In this sample: p-fmatter-001.xhtml

[Notes]

- For the sake of convenience, we will define all the pages between the cover page and the title page as front matter..
- These pages may not necessarily contain images.

-----[sample code]-----

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops"
  xml:lang="ja"
  class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-image">
<div class="main">
<p></p>
</div>
</body>
</html>
```

■ Title Page

[filename: p-titlepage.xhtml]

[Notes]

- Content and text orientation vary depending on the publisher and the work.
(All content listed below, such as class names, are for reference purposes only.)

-----[sample code]-----

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops"
  xml:lang="ja"
  class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-titlepage">
<div class="main">
<div class="book-title">
<div class="book-title-before">
<p>subtitle - before</p>
</div>
<div class="book-title-main">
<p>main title</p>
</div>
<div class="book-title-after">
<p>sub title - after</p>
</div>
</div>
```

```

<div class="author">
<p>author name 1</p>
<p>author name 2</p>
</div>
<div class="label">
<p class="label-logo"></p>
<p class="label-name">●● Bunko</p>
</div>
</div>
</body>
</html>

```

■Precautions page for electronic versions

[filename: p-caution.xhtml]

[Notes]

- Content, text orientation, insertion positions, etc. vary depending on the publisher and the work.
- Since we assume this will primarily be used to list "standard precautions including layout information such as text size, to be inserted in all works," use a normal body page unless otherwise instructed for precautions that differ for each work.

```

-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="vrtl"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-caution">
<div class="main">
<p>All rights reserved, etc.</p>
</div>
</body>
</html>

```

■ Contents Page

[filename: p-toc.xhtml]

[Notes]

- This page is unnecessary if using a navigation document as the contents of a work's body as well.
 - Content and text orientation vary depending on the work.
 - Always include an id at the jump destination unless otherwise instructed.
 - Do not set a link to return to the contents from the jump destination unless otherwise instructed.
 - Classes used for heading elements and the like are not fixed.
- (You can directly designate the class of sizes, fonts and the like, as in `<h1 class="gfont font-1em30">.`)

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="vrtl"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-toc">
<div class="main">
<h1 class="mokuji-midashi">□ contents heading</h1>
<p><br/></p>
<p><br/></p>
<p><a href="p-001.xhtml#toc-001">contents item 1</a></p>
<p>□<a href="p-002.xhtml#toc-002"><span class="font-0em80"> contents item 2</span></a></p>
<p>□<a href="p-002.xhtml#toc-003"><span class="font-0em80"> contents item 3</span></a></p>
</div>
</body>
</html>
```

■ Title Page

[filename: p-***.xhtml] *In the example: "p-001.xhtml"

[Notes]

- Content and text orientation vary depending on the work.
 - Regarding the designated position of ids receiving links from the contents (same for body pages): Designate ids to heading-like strings of text identical to the contents item, unless otherwise instructed. If there are no heading-like elements (only images, etc.), or if there is a heading, but there is content immediately preceding it that should also be displayed, attach the id to the element (block level element) designated with `{ display: block; }` in the CSS, such as `<p>` or `<div>` in the near vicinity of the area containing the jump destination.
 - When using 3-digit serial numbers in file names, such as "p-001", be sure to adjust the file names at an appropriate time if they exceed three digits.
 - Classes used for heading elements and the like are not fixed.
- (You can directly designate the class of sizes, fonts and the like, as in `<p class="gfont font-1em50">.`)

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="vrtl"
>
```

```

<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-tobira">
<div class="main">
<p class="tobira-midashi" id="toc-001">Chapter 1 □abcdefghijklmnopqrstuvxyz</p>
</div>
</body>
</html>

```

■Body Page (Vertical Orientation)

[filename: p-***.xhtml] *In the example: "p-002.xhtml"

[Notes]

- Content and text orientation vary depending on the work.
- Classes used for heading elements and the like are not fixed.

(You can directly designate the class of sizes, fonts and the like, as in <h1 class="gfont font-1em30">.)

-----[sample code]-----

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="vrtl"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-text">
<div class="main">
<h1 class="oo-midashi" id="toc-002"> Section 1 □abcdefghijklmnopqrstuvxyz</h1>
<p><br/></p>
<p><br/></p>
<p>□ This is a sample sentence.</p>
<p>This is a sample sentence.</p>
<p><br/></p>
<h2 class="ko-midashi" id="toc-003">□□□□Item 1</h2>
<p><br/></p>
<p>□ This is a sample sentence.</p>
<p>□ This is a sample sentence.</p>
<p></p>
<p>□ This is a sample sentence.</p>
</div>
</body>
</html>

```

■Body Page (Horizontal Orientation)

[filename: p-***.xhtml] *In the example: "p-003.xhtml"

[Notes]

- This is basically the same as the vertical orientation, except the <html> class has changed.
- Content and text orientation vary depending on the work.

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-text">
<div class="main">
<p><br/></p>
<p><br/></p>
<p>□ This is a sample sentence.</p>
<p>This is a sample sentence.</p>
</div>
</body>
</html>
-----
```

■Body Page (Image Only)

[filename: p-***.xhtml] *In the example: "p-004.xhtml"

[Notes]

- Content and text orientation vary depending on the work.
- The default is horizontally centered text beginning from the top of the page.
- If it is ok to align the text to the right, the <html> class can be the same designation as in the vertical orientation.
- Do not list anything other than image display designations.

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-image">
<div class="main">
<p></p>
</div>
</body>
</html>
-----
```

■ Designating the Overall Page Position Alignment

[filename: p-***.xhtml] *In the example: "p-005.xhtml"

[Notes]

- Text pages and image pages are handled the same.
- Align the position of all pages Designate class="align-***" in <div class="main">.

align-justify: Align text at both ends (Only the end of the line is aligned at the start. This is the default for body text.)

align-start: Align content at start (Note that since image pages are horizontally oriented, start = align content to left.)

align-left: Align content at start

align-center: Align text in center (Centers content between top and bottom when vertically oriented. Centers text between left and right when horizontally oriented. This is the default for image pages.)

align-end: Align content at end

align-right: Align content at end

- The example aligns image pages to the left.

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-image">
<div class="main align-left">
<p></p>
</div>
</body>
</html>
```

■ Imprint Page

[filename: p-colophon.xhtml]

[Notes]

- Content and text orientation vary depending on the publisher and the work.
- (All of the following content, including class names, is for reference purposes only.)

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-colophon">
<div class="main">
<div class="book-title">
<div class="book-title-before">
<p>subtitle - before</p>
```

```

</div>
<div class="book-title-main">
<p>subtitle</p>
</div>
<div class="book-title-after">
<p>subtitle - after</p>
</div>
</div>
<div class="author">
<p>author name 1</p>
<p>author name 2</p>
</div>
<div class="label">
<p class="label-logo"></p>
</div>
<div class="release-date">
<p>Published: □MM/DD/YYYY</p>
</div>
<div class="publisher-data">
<p class="publish-person">Publisher:□●●●●</p>
<p class="publish-company">Office□●●Publishing, Co., Ltd.</p>
<p class="publish-address">●●, ●●-ku, Tokyo□000-0000</p>
<p class="publish-url">http://www.***.co.jp/</p>
</div>
<div class="copyright">
<p>(C) author01 20xx</p>
<p>(C) author02 20xx</p>
</div>
<div class="kotowarigaki">
<p>(Place any prohibitions in the imprint here.)</p>
</div>
<div class="original-books">
<p>This electronic publication was created based on the following:</p>
<p class="original-first-edition">●● Bunko "Original Draft Title" First printed: MM/DD/YYYY</p>
<p class="original-used-edition">XXth edition published: MM/DD/YYYY</p>
</div>
</div>
</body>
</html>

```

■Advertisement Page

[filename: p-ad-***.xhtml] *In the example: "p-ad-001.xhtml"

[Notes]

- Content and text orientation vary depending on the publisher and the work.
- Not necessarily an image page.

```

-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>

```

```

<body class="p-image">
<div class="main">
<p></p>
</div>
</body>
</html>

```

[Reference Information] *Not recommended by this guide.

■ Vertically oriented pages centered horizontally

[filename: p-***.xhtml]

[Notes]

- Not all RSs support the combination of different text orientations, so be sure to check the performance of the RS you plan to use.
- If the content runs off the page, part of it will not be shown, so be sure to check things like the size of the screen you plan to display the content on.
- In order to make nests out of vertical blocks in horizontally oriented text, watch out for class values (especially margins, etc.) that are not overwritten when changing the direction of the orientation.
- The top and bottom margins will be reduced to 0, so be sure to set < div > in addition within <div class="main"> that designates margins or padding as necessary. (This method disapproves the addition of margins and padding in the <body> and <div class="main">. Be aware that the display would not always become as expected even when using classes designated in the <body> , such as .p-text designating margins and so on.)
- If you want to designate "vrtl block-align-center2 in <div class="main"> rather than in the <body> on text pages, be sure to set the margins and padding to 0.
- To align text to the left, change "block-align-center" below to "block-align-left (or start)".

*Since this is a horizontally oriented page, remember that "end=right".

- This usage method is the same for title pages (.p-tobira), etc.
- When designating vertical middle of screen in a horizontally oriented page, replace "hltr" and "vrtl" in the sample code below.

Designate width-100per to main, otherwise the text would be aligned to the right.

※The base is vertically oriented, so the elements are placed from right to left.

-----[sample code]-----

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-text">
<div class="main vrtl block-align-center">
<div class="start-2em">      // ←To lower the whole text on the page, put another <div> in the inside
<p>abcdefghijklmnopqrstuvwxy</p>
</div>
</div>
</body>
</html>

```

*The following example places images in the lower-left corner.

[Notes]

- When aligning content with the bottom of the page, be aware that the height of `<div class="main">` directly under the body must be set to 100%, or their positions will be shifted when using WebKit products.

※When using horizontal orientation, the width should be set to 100%, not the height.

- In the following example, the image has been reduced in screen height by 50% in order to clearly demonstrate the effect

(the ``).

-----[sample code]-----

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
class="hltr"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/book-style.css"/>
</head>
<body class="p-text">
<div class="main vrtl block-align-left height-100per">

<div class="align-end">      // ←To align the whole text to the end, put another <div> in the inside
<p></p>
</div>

</div>
</body>
</html>
```

B. Fixed Layouts

Files required for settings

■ Colors in templates

Gray: Sections common to all works (in principle, do not change)

Blue: Sections that are common to all works, but change for each work

Red: Sections of the work that use that template, which require special attention (in principle, do not change)

Black: Non-standard parts (vary for each work, publisher)

■ mimetype

[filename: mimetype]

[Notes]

- Same as reflowable

-----[sample code]-----
application/epub+zip

■ container.xml in META-INF

[filename: container.xml]

[Notes]

- Same as reflowable

-----[sample code]-----
<?xml version="1.0"?>
<container
version="1.0"
xmlns="urn:oasis:names:tc:opendocument:xmlns:container"
>
<rootfiles>
<rootfile
full-path="item/standard.opf"
media-type="application/oebps-package+xml"
>
</rootfiles>
</container>

■ Navigation Documents

[filename: navigation-documents.xhtml]

[Notes]

- Basically the same as reflowable.
- When file names are sequential, adjust them at appropriate times (the contents in the example below is p-001.xhtml.)

-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
>
<head>
<meta charset="UTF-8"/>
<title>Navigation</title>
</head>

```

<body>
<nav epub:type="toc" id="toc">
<h1>Navigation</h1>
<ol>
<li><a href="xhtml/p-cover.xhtml">Cover</a></li>
<li><a href="xhtml/p-001.xhtml">Contents</a></li>
<li><a href="xhtml/p-colophon.xhtml">imprint</a></li>
</ol>
</nav>
</body>
</html>

```

■ OPF Files

[filename: standard.opf]

[Notes]

- The following items differ from reflowable:

A prefix line has been added to the <package> element.

Two <meta> elements have been added to the <!-- Fixed-Layout Documents designation --> part.

The only style sheet is fixed-layout-jp.css.

properties="rendition:page-spread-center" has been added to the cover page in <itemref>, in the <spine> element.

In the <spine> element's <itemref>, the left and right pages should be paired together except for the cover image.

*All else is the same as reflowable.

- When the idref values in the <spine> element's <itemref> overlap, errors may occur such as the screen going blank (Readium) or the page loops (Firefox's EPUBReader). If you want to show the same image multiple times, we recommend you prepare another xhtml file to call up the image just to be safe (for a second white image, white2.xhtml, etc.).

```

-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<package
xmlns="http://www.idpf.org/2007/opf"
version="3.0"
xml:lang="ja"
unique-identifier="unique-id"
prefix="rendition: http://www.idpf.org/vocab/rendition/#
      ebpaj: http://www.ebpaj.jp/
      dpfj: https://www.dpfj.or.jp/"
>
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
<!-- work title -->
<dc:title id="title">work title 1</dc:title>
<!-- author name -->
<dc:creator id="creator01">author name 1</dc:creator>
<meta refines="#creator01" property="role" scheme="marc:relators">aut</meta>
<dc:creator id="creator02">author name 2</dc:creator>
<meta refines="#creator02" property="role" scheme="marc:relators">aut</meta>
<!-- publisher name -->
<dc:publisher id="publisher">publisher name</dc:publisher>
<!-- language -->
<dc:language>ja</dc:language>
<!-- file id -->
<dc:identifier id="unique-id">urn:uuid:78962f36-6855-4455-8c90-ae787e99cc2b</dc:identifier>
<!-- Revision Date -->
<meta property="dcterms:modified">2025-06-01T00:00:00Z</meta>
<!-- Fixed-Layout Documents designation -->
<meta property="rendition:layout">pre-paginated</meta>
<meta property="rendition:spread">landscape</meta>
<!-- etc. -->
<meta property="ebpaj:guide-version">1.1.4</meta>
<meta property="dpfj:guide-version">1.1.4</meta>

```

```

</metadata>
<manifest>
<!-- navigation -->
<item media-type="application/xhtml+xml" id="toc" href="navigation-documents.xhtml"
properties="nav"/>
<!-- style -->
<item media-type="text/css" id="fixed-layout-jp" href="style/fixed-layout-jp.css"/>
<!-- image -->
<item media-type="image/jpeg" id="cover" href="image/cover.jpg" properties="cover-image"/>
<item media-type="image/jpeg" id="i-white" href="image/i-white.jpg"/>
<item media-type="image/jpeg" id="i-001" href="image/i-001.jpg"/>
<item media-type="image/jpeg" id="i-002" href="image/i-002.jpg"/>
<item media-type="image/jpeg" id="i-003" href="image/i-003.jpg"/>
<item media-type="image/jpeg" id="i-004" href="image/i-004.jpg"/>
<item media-type="image/jpeg" id="i-005" href="image/i-005.jpg"/>
<item media-type="image/jpeg" id="i-colophon" href="image/i-colophon.jpg"/>
<!-- xhtml -->
<item media-type="application/xhtml+xml" id="p-cover" href="xhtml/p-cover.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-white" href="xhtml/p-white.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-001" href="xhtml/p-001.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-002" href="xhtml/p-002.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-003" href="xhtml/p-003.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-004" href="xhtml/p-004.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-005" href="xhtml/p-005.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-colophon" href="xhtml/p-colophon.xhtml"
properties="svg"/>
<item media-type="application/xhtml+xml" id="p-white2" href="xhtml/p-white2.xhtml"
properties="svg"/>
</manifest>
<spine page-progression-direction="rtl">
<itemref linear="yes" idref="p-cover" properties="rendition:page-spread-center"/>
<itemref linear="yes" idref="p-white" properties="page-spread-right"/>
<itemref linear="yes" idref="p-001" properties="page-spread-left"/>
<itemref linear="yes" idref="p-002" properties="page-spread-right"/>
<itemref linear="yes" idref="p-003" properties="page-spread-left"/>
<itemref linear="yes" idref="p-004" properties="page-spread-right"/>
<itemref linear="yes" idref="p-005" properties="page-spread-left"/>
<itemref linear="yes" idref="p-colophon" properties="page-spread-right"/>
<itemref linear="yes" idref="p-white2" properties="page-spread-left"/>
</spine>
</package>

```

When describing a standard size and fallbacks for RSs that do not yet support SVG to an OPF file to quickly display a page-fit fixed-size image on all pages, we recommend describing it as follows.

[filename: standard.opf]

[Notes]

- We added a prefix declaration for fixed layouts to the package element.
- We added a standard size designation for the page image to the <metadata> element.
- We made the image measurements all the same size, and entered the number of vertical and horizontal pixels of the full-sized image in the area highlighted in blue below.
- An image has been designated to act as a fallback in the <item>'s xhtml file, out of consideration for RSs that do not yet support SVG.

```

-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<package
xmlns="http://www.idpf.org/2007/opf"
version="3.0"
xml:lang="ja"
unique-identifier="unique-id"
prefix="rendition: http://www.idpf.org/vocab/rendition/#
ebpaj: http://www.ebpaj.jp/
fixed-layout-jp: http://www.digital-comic.jp/"
>
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
(*Omitted)
<!-- Fixed-Layout Documents designation -->
<meta property="rendition:layout">pre-paginated</meta>
<meta property="rendition:spread">landscape</meta>
<!-- standard size -->
<meta property="fixed-layout-jp:viewport">width=848, height=1200</meta>
<!-- etc. -->
<meta property="ebpaj:guide-version">1.1.4</meta>
<meta property="dpfj:guide-version">1.1.4</meta>
</metadata>
<manifest>
(*Omitted)
<!-- xhtml -->
<item media-type="application/xhtml+xml" id="p-cover" href="xhtml/p-cover.xhtml" properties="svg"
fallback="cover"/>
<item media-type="application/xhtml+xml" id="p-white" href="xhtml/p-white.xhtml" properties="svg"
fallback="i-white"/>
<item media-type="application/xhtml+xml" id="p-001" href="xhtml/p-001.xhtml" properties="svg"
fallback="i-001"/>
<item media-type="application/xhtml+xml" id="p-002" href="xhtml/p-002.xhtml" properties="svg"
fallback="i-002"/>
<item media-type="application/xhtml+xml" id="p-003" href="xhtml/p-003.xhtml" properties="svg"
fallback="i-003"/>
<item media-type="application/xhtml+xml" id="p-004" href="xhtml/p-004.xhtml" properties="svg"
fallback="i-004"/>
<item media-type="application/xhtml+xml" id="p-005" href="xhtml/p-005.xhtml" properties="svg"
fallback="i-005"/>
<item media-type="application/xhtml+xml" id="p-colophon" href="xhtml/p-colophon.xhtml" properties="svg"
fallback="i-colophon"/>
<item media-type="application/xhtml+xml" id="p-white2" href="xhtml/p-white2.xhtml" properties="svg"
fallback="i-white"/>
</manifest>
(Remainder omitted.)
-----

```

XHTML Document Files

■ Colors in templates

Gray: Sections common to all works (in principle, do not change)

Blue: Sections that are common to all works, but change for each work

Red: Sections of the work that use that template, which require special attention (in principle, do not change)

Black: Non-standard parts (vary for each work, publisher)

■ Cover Page

[filename: p-cover.xhtml]

[Notes]

- List the image's original size in the following three locations with blue text.
- Unify all image sizes within the work.

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops"
  xml:lang="ja"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/fixed-layout-jp.css"/>
<meta name="viewport" content="width=848, height=1200"/>
</head>
<body epub:type="cover">
<div class="main">
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  width="100%" height="100%" viewBox="0 0 848 1200">
<image width="848" height="1200" xlink:href="../image/cover.jpg"/>
</svg>
</div>
</body>
</html>
-----
```

■ Body Page

[filename: p-***.xhtml] *In the example: "p-002.xhtml"

[Notes]

- Except for the absence of epub:type="cover", this is identical to the Cover Page.

```
-----[sample code]-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:epub="http://www.idpf.org/2007/ops"
  xml:lang="ja"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/fixed-layout-jp.css"/>
<meta name="viewport" content="width=848, height=1200"/>
</head>
<body>
```

```

<div class="main">
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
width="100%" height="100%" viewBox="0 0 848 1200">
<image width="848" height="1200" xlink:href="../image/i-002.jpg"/>
</svg>
</div>
</body>
</html>

```

■Body Image Map (Clickable Map) Page

[filename: p-***.xhtml] *In the example: "p-001.xhtml"

[Notes]

- List the link destination file name in a element's xlink:href attribute.
- List the starting coordinates (upper left) for the clickable area in the rect element's x and y attributes.
- List the size of the clickable area in the rect element's width and height attributes.
- Not all current RSs can support this, and some RSs restrict the number of links. Be sure to check the performance and actions of the RS you plan to use in advance.

-----[sample code]-----

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html
xmlns="http://www.w3.org/1999/xhtml"
xmlns:epub="http://www.idpf.org/2007/ops"
xml:lang="ja"
>
<head>
<meta charset="UTF-8"/>
<title>work title</title>
<link rel="stylesheet" type="text/css" href="../style/fixed-layout-jp.css"/>
<meta name="viewport" content="width=848, height=1200"/>
</head>
<body>
<div class="main">
<svg xmlns="http://www.w3.org/2000/svg" version="1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
width="100%" height="100%" viewBox="0 0 848 1200">
<image width="848" height="1200" xlink:href="../image/i-001.jpg"/>
<a xlink:href="p-002.xhtml" target="_top"><rect fill-opacity="0.0" x="476" y="1000" width="300"
height="60"/></a>
<a xlink:href="p-colophon.xhtml" target="_top"><rect fill-opacity="0.0" x="476" y="1075" width="300"
height="60"/></a>
</svg>
</div>
</body>
</html>

```

List of Forms

■ Formal Paragraphs

[Notes]

Surround all formal paragraphs with `<p></p>`.

(The margins and padding for `<p>` in the style sheet are 0.)

[HTML elements]

`<p></p>`: Formal Paragraph

-----[sample code]-----

`<p>Text</p>`

`<p>Text</p>`

■ Line Breaks, Blank Lines

[Notes]

As a general rule, `<p>
<p>` is used for blank lines.

[HTML elements]

`<p>
</p>`: Blank line

Ex: 1 blank line

-----[sample code]-----

`<p>Text</p>`

`<p>
</p>`

`<p>Text</p>`

■ Horizontal text in vertically oriented pages

[Notes]

This guide does not assume superscripts/subscripts or text decorations for designated horizontal text in vertically oriented pages. These must be made into images.

When designating decorative text such as fonts, begin the designation from outside the horizontal text, as if surrounding it.

[CSS class]

`class="tcy"`: 1-3 digits. 4 or more digits must be made into an image. (See the section on non-standard kanji images for more information on how to insert these images.)

Ex: 2-digit horizontal text in a vertically oriented page

-----[sample code]-----

`99`

■ Text direction when vertically oriented

[CSS class]

`class="sideways"`: Sideways

`class="upright"`: Upright (full-width)

`class="upright-1"`: Upright (half-width) *text will not be centered, so it requires a separate designation.

Ex: Sideways

-----[sample code]-----

`10 ～ 20`

Ex: Upright (full-width)

-----[sample code]-----

`"—"` (minus sign)

Ex: Upright (half-width)

-----[sample code]-----

The letter ` É ` is used in French.

■Rubies (Phonetic readings of kanji printed in smaller sized text next to the kanji)

[HTML elements]

`<ruby>Kanji<rt>Ruby string</rt></ruby>`

Ex: Group Ruby

-----[sample code]-----
`<ruby>紫陽花<rt>あじさい(AJISAI)</rt></ruby>`

Ex: Mono Ruby

-----[sample code]-----
`<ruby>月<rt>TSUKI</rt></ruby>`

Ex: Kanji Combination Ruby

-----[sample code]-----
`<ruby>太<rt>TAI</rt>陽<rt>YOU</rt></ruby>`

[Notes]

- Split up group rubies that are too long.

If rubies are longer than a single line, some RSs omit text or wrap them into the same line rather than the next line. Make sure rubies are not too long when using them to spell out the pronunciation or meaning of foreign sentences.

Also note that some RSs prevent words designated as group rubies from being split up.

In such cases, the RS sends the word to the next line, leaving a blank space in the previous line, or, if the `<body>` is justified, equally sized spaces will appear between each character in the line.

If a group ruby string (`<rt>〜</rt>`) is too long, separate the ruby at an appropriate place while checking it on the screen, and leave a note when delivering the work for the first proofreading.

Unless otherwise instructed, the person in charge of creating the file shall be responsible for ruby lengths and separation locations. The presence of spaces between characters in the previous line can be tolerated to a certain extent.

Ex: A ruby that is too long

-----[sample code]-----
`<ruby>未四曾有三一事不レ被二無常吞一<rt>いまだかつていちじのむじょうにのまれざるわあらず(IMADAKATSUTEICHIJINOMUJOUNINOMAREZARUWAARAZU)</rt></ruby>`

↓

-----[sample code]-----
`<ruby>未四曾有三<rt>いまだかつて</rt></ruby><ruby>一事不レ被二<rt>いちじのむじょうに</rt></ruby><ruby>無常吞一<rt>のまれざるわあらず(NOMAREZARUWAARAZU)</rt></ruby>`

■Designating Special Text

- Text references, numerical references

[Notes]

If using character references (character entity references), only the five characters listed below are allowed in the XML specifications.

However, the characters `["]` and `[']` normally use simple text characters in the body, rather than references.

All others use numerical references (numerical character references) (`[©]`, etc.).

`&` : `&`;

< : <
 > : >
 " : "
 ' : '

■Text Decoration

[CSS class]

Font

class="mfont": Ming-cho

class="gfont": Gothic

Text size (*Check the CSS file for usable values)

class="font-0em80" : Designate this with the number of characters (em). (0.8em in this example.)

class="font-080per": Designate this with the enlargement/reduction rate (%). (80% in this example.)

Bold

class="bold": Make text bold

class="font-weight-normal": Remove bold

Italic

class="italic": Italicize text

class="font-style-normal": Remove italic

Ex: Font

-----[sample code]-----

Text

Text

Ex: Text Size

-----[sample code]-----

Text

Text

Ex: Bold

-----[sample code]-----

Text

Ex: Italic

-----[sample code]-----

Text

Ex: Designating text that runs on to the next page

-----[sample code]-----

<div class="gfont">

<p>Text</p>

<p>Text</p>

</div>

■Emphasizing/Negating Text

- Emphatic Dots

[CSS class]

class="em-sesame" : Punctuation mark

class="em-sesame-open" : Punctuation mark (hollow)

class="em-dot" : Dot

class="em-dot-open" : Dot (hollow)

class="em-circle" : Circle

class="em-circle-open" : Circle (hollow)

class="em-double-circle" : Double circle

class="em-double-circle-open" : Double circle (hollow)

class="em-triangle" : Triangle

class="em-triangle-open" :Triangle (hollow)

Ex: Sesame mark

```
-----[sample code]-----
<span class="em-sesame">Text</span>
-----
```

• Emphatic Lines

[CSS class]

class="em-line" : Line along the right side of text when vertically oriented, under the text when horizontally oriented
 class="em-line-outside" : Line along the left side of text when vertically oriented, above the text when horizontally oriented

Ex: Right/under line

```
-----[sample code]-----
<span class="em-line">Text</span>
-----
```

• Negation Lines

[Notes]

Things like double negation lines must be made as non-standard kanji images.

[CSS class]

class="line-through" : Negation line (a single line drawn through the text, scratching it out)

Ex:

```
-----[sample code]-----
<span class="line-through">Text</span>
-----
```

■ Images

[Notes]

Designate the class in .

Be sure to insert [alt=""], unless instructed otherwise by someone like the publisher, in accordance with HTML5's alt rules.

[HTML elements]

```

```

• Normal Images

Ex: In-line image

```
-----[sample code]-----
This sentence is a sample.
-----
```

• Images of Non-standard Kanji Characters

[Notes]

- Any characters other than those listed in JIS X 0213:2004 are treated as non-standard kanji images.
- Follow the publisher's instructions when deciding which font to base the character shapes on.
- Create a transparent background and use a transparent PNG saved in 8bit PNG.

*Keep in mind that some RSs allow readers to change the background color.

[CSS class]

class="gaiji" : an image the size of 1 character in height and width (example size of created image: 128px×128px)

class="gaiji-line" : Vertically stretched character with a width the size of 1 character (example size of created image:

128px wide × desired height)

class="gaiji-wide" : Horizontally stretched character with a height the size of 1 character (example size of created image: 128px high × desired width)

Ex: size of 1 character

-----[sample code]-----
This sentence is a sample.

Ex: Vertically stretched character with a width the size of 1 character

-----[sample code]-----
This sentence is a sample.

Ex: Horizontally stretched character with a height the size of 1 character

-----[sample code]-----
This sentence is a sample.

• Designating Image Sizes

[Notes]

Sizes are designated by the number of characters, or by the image's ratio to the screen.

For more information, see the section entitled "Size" later in this guide.

In general, designate the size of images that appear in-line (within a line) by the number of characters.

Refer to CSS to learn how many characters can be designated.

Note that large images may be cut off the screen or change shape if not designated with "max-".

[CSS class]

height-*em : Designate the image's height by the number of characters (ex: "height-2em50".....image height is 2.5 characters high)

height-***per : Designate the image's height by its ratio to the screen height

(ex: "height-010per".....image height is 10% the height of the screen)

max-height-*em : Designate the image's maximum height by the number of characters

max-height-***per : Designate the image's maximum height by its ratio to the screen height

width-*em : Designate the image's width by the number of characters

width-***per : Designate the image's width by its ratio to the screen width

max-width-*em : Designate the image's maximum width by the number of characters

max-width-***per : Designate the image's maximum width by its ratio to the screen width

max-size-*em : Designate the image's maximum height and width by the number of characters

max-size-***per : Designate the image's maximum height and width by its ratio to the screen height and width

*About usable values (such as auto, none, 0, etc.), refer to "■ Handling Boxes"→"• Size"

Ex: Reduce the size of an image and display it within a line

-----[sample code]-----
This sentence is a sample.

• Fitting Images to Pages

[Notes]

This adjusts image sizes to fit them to the screen size. Note, however, that images cannot be enlarged beyond their original sizes.

Designate the maximum value of page fit image sizes with a "max-" attached to a "%".

Images cannot be fit to pages by designating the number of characters.

[CSS class]

class="fit" : Designate an image to be fit to the page

Ex: 100% fit to page height or width

-----[sample code]-----
<p></p>

 Ex: Always display images at 50% the size of the page height

-----[sample code]-----
`<p></p>`

 Ex: Always display images at the size of 20em (do not fit images to pages)

-----[sample code]-----
`<p></p>`

■Headings

[Notes]

Unless otherwise instructed, use the following heading elements in the heading's main title part, and use `<p></p>` for neighboring related lines such as subtitles.

Regarding heading-level settings, follow the publisher's instructions.

If no instructions are provided, the creator shall be responsible for heading-level settings.

In decoration settings such as font and size, treat heading elements as the same kind of block elements as `<div>`.

Be aware, however, that `<div>` and `<p>` cannot be used in heading elements.

When aligning, lowering, or indenting text, surround heading elements with `<div>`.

[HTML elements]

`<h1>` - `<h6>`

Ex: Decorating heading text

-----[sample code]-----
`<h1 class="gfont font-1em30">Chapter 1</h1>`

Ex: Hanging indent in heading

-----[sample code]-----
`<div class="h-indent-4em">`
`<h1>Chapter 1 Heading</h1>`
`</div>`

· Window classes for headings

[Notes]

The CSS file contains the following classes exclusively for headings.

You can also make stylistic settings, such as font and size, to accommodate each work.

Refer to the section entitled "Default CSS Files" later in this guide to learn more about customizing style sheets.

[CSS class]

`class="mokuji-midashi"`

`class="tobira-midashi"`

`class="oo-midashi"`

`class="naka-midashi"`

`class="ko-midashi"`

Ex: Setting used for heading class

-----[sample code]-----
`<h1 class="oo-midashi">Chapter 1</h1>`

· Heading Images

[Notes]

When images are used as headings, handle them the same as you would simple in-line or page-fit images.

You can enclose the images with heading elements (`<h1>` - `<h6>`).

Insert the headline text contained in the images in alt, to facilitate readings and searches.

Ex: The basic form of a heading image

-----[sample code]-----
 <p></p>

■Aligning the positions of in-line elements

• Changing the baseline

[Notes]

In vertically oriented pages, the correct specification for the vertical-align's baseline is the center of text and images. Be aware that the tops and bottoms of lines may be interpreted differently depending on the RS.

[CSS class]

class="vertical-align" : Succeeds the parent element's vertical-align

class="vertical-align-baseline" : Displays the child element's baseline aligned with the parent element's baseline.

class="vertical-align-middle" : Displays the middle of the child element aligned with the parent element's middle baseline.

class="vertical-align-top" : Displays the top of the child element aligned with the top of the line.

class="vertical-align-bottom" : Displays the bottom of the child element aligned with the bottom of the line.

class="vertical-align-text-top" : Displays the top of the child element aligned with the top of the parent element's text.

class="vertical-align-text-bottom" : Displays the bottom of the child element aligned with the bottom of the parent element's text.

class="vertical-align-sub" : Shows the subscript position of the parent element.

class="vertical-align-super" : Shows the superscript position of the parent element.

Ex: Display the middle of the text aligned with the parent element's middle baseline (the height of the middle of the lower-case letter "x").

-----[sample code]-----
 Textx XxText

• Superscript and Subscript

[Notes]

The font size is set to "smaller".

*Note that [class="vertical-align-super"] and [class="vertical-align-sub"] do not change the text size.

[CSS class]

class="super"

class="sub"

Ex: Superscript (western text)

-----[sample code]-----
 32

Ex: Superscript (Japanese, full-width text)

-----[sample code]-----
 H 2 O

Ex: Subscript (western text)

-----[sample code]-----
 H2O

Ex: Subscript (Japanese, full-width text)

-----[sample code]-----
 A B C

- **Kunten** (guiding marks for rendering Chinese into Japanese)

[Notes]

Do not use the small kunten characters in Unicode text. Instead, designate regular-sized text.

[CSS class]

```
class="kunten"
class="kunten-okuri"
```

Ex: ㄥ and 一 marks, etc. (Marks showing the order in which to read characters in Chinese texts so that Japanese readers can understand their meaning.) (When vertically oriented, display in a small size on the left side of the text within the line.)

```
-----[sample code]-----
<span class="kunten">ㄥ</span>
<span class="kunten">一</span>
```

Ex: Okurigana (The declensional kana ending of a word beginning with kanji.) (When vertically oriented, display in a small size on the right side of the text within the line.)

```
-----[sample code]-----
saki<span class="kunten-okuri">nzureba</span>sunawa<span class="kunten-okuri"> chi</span>
```

- **Small Script Text**

[Notes]

The interpretation of vertical-align's "text-top" when the text is vertically oriented differs based on the RS, so we have provisionally set "vertical-align" to "super" for the vertical orientation.

[CSS class]

```
class="kogaki"
```

Ex:

```
-----[sample code]-----
furukeyakawazutobi<span class="kogaki">n</span>darumizunooto
```

■Aligning Lines

[Notes]

Unless provided with instructions about forming lines, always place a line break in the source immediately after both <div> and </div>.

The <body>'s text-align has been justified, so the end of the lines line up with the side of the screen.

However, depending on the screen size large spaces may open up between English words and the like, making sentences awkward to read.

Basically leave the alignment as-is, but change it to align-start later if instructed.

[CSS class]

```
class="align-start" : Align text at the start of the line (Top for vertically oriented text, left side for horizontally oriented)
class="align-center" : Center text
class="align-end" : Align text at the end of the line (Bottom for vertically oriented text, right side for horizontally oriented)
class="align-justify" : Align text at the start of the line (Automatically adjust text spacing so the end of the lines align)
```

Ex: Align text at the end of the line

```
-----[sample code]-----
<div class="align-end">
<p>Text</p>
<p>Text</p>
</div>
```

Ex: Detering justification (changing the alignment from both ends to only the start of lines)

```
-----[sample code]-----
<p>This is a pen.</p>
-----
↓
-----[sample code]-----
<div class="align-start">
<p>This is a pen.</p>
</div>
-----
```

■ Lowering Text and Indenting



















• Lowering text using full-width spaces

[Notes]

Since small screens wrap text, you should replace strings of 6 or more characters to lower with designations at the start of the line.

Note that page-fit images (class="fit") cannot be lowered with full-width spaces.

Ex: Use full-width spaces to lower all content by 4 characters *.....full-width space

```
-----[sample code]-----
<h1>Heading Text</h1>
<p>Text</p>
<p>src="..image/img-001.jpg" alt=""/></p>
<p>class="height-3em" src="..image/img-001.jpg" alt=""/></p>
-----
```

• Lowering text by designating the class

[Notes]

This lowers text from the start of the line. When horizontally oriented, it will be lowered from the left side.

When vertically oriented, from the top. As a general rule, designate "start-*em" and the number of characters to lower in the <div> class.

Note that CSS uses margins to adjust positions, so be sure to avoid designating characters to lower at the same time as designating blank margins in the same element.

[CSS class]

class="start-*em"

Ex: Lower all content by 4 characters from the start of the line

```
-----[sample code]-----
<div class="start-4em">
<h1>HeadingText</h1>
<p>Text</p>
<p></p>
<p></p>
</div>
-----
```

Ex: Nest lowering text

```
-----[sample code]-----
<div class="start-4em">
<p>Text</p>
<div class="start-2em">6 characters will be lowered overall
<p>Text</p>
</div>
</div>
-----
```


• Raising Text

[Notes]

This is basically the same process as lowering text. This raises text from the end of the line.

When horizontally oriented, it will be raised from the right side.

When vertically oriented, from the bottom. As with lowering text, avoid designating text to raise at the same time as designating blank margins in the same element.

[CSS class]

class="end-*em"

Ex: Raise all content 4 characters from the end of the line

```
-----[sample code]-----
<div class="end-4em">
<p>□Peter Piper picked a peck of pickled peppers.</p>
</div>
-----
```

• Indenting the start of a line

[Notes]

This indents the start of a line. This differs from lowering text in that only the beginning of the line is lowered.

When horizontally oriented, text is lowered from the left side. When vertically oriented, from the top.

[CSS class]

class="indent-*em" : Designate the number of characters to indent from the start of the line

Ex: Indent a line by 1 character from the start of the line

```
-----[sample code]-----
<div class="indent-1em">
<p>Text</p>
<p>Text</p>
</div>
-----
```

• Protruding Indentations (Hanging Indentations)

[Notes]

This designates the start position of lines that have wrapped around from the side of the screen. When horizontally oriented, the text is lowered from the left. When vertically oriented, from the top.

"h-" stands for "hanging."

Depending on the width, the spacing of characters may not line up precisely when horizontally oriented.

Note that CSS uses the negative values of indent and padding to adjust positions, so be sure to avoid designating protruding indentations at the same time as designating blank padding in the same element.

[CSS class]

class="h-indent-*em" : Uses the number of characters to designate the start position of wraparound text from the start of the line

Ex: Display the start of a wraparound line aligned with the third character in the initial line.

```
-----[sample code]-----
<div class="h-indent-2em">
<p>Man: Peter Piper picked a peck of pickled peppers.</p>
<p>Woman: A peck of pickled peppers, Peter Piper picked.</p>
</div>
-----
```

*When the text wraps around, it should appear as follows:

```
Man: Peter Piper picked a peck of
    Pickled peppers.
Woman: A peck of pickled peppers,
```

Peter Piper picked.

Ex: Combining with lowered text

```
-----[sample code]-----
<div class="start-2em">
<div class="h-indent-2em">
<p>Man: Peter Piper picked a peck of pickled peppers.</p>
</div>
</div>
```

Also:

```
-----[sample code]-----
<div class="start-2em h-indent-2em">
<p>Man: Peter Piper picked a peck of pickled peppers.</p>
</div>
```

Ex: Using together with blank space (padding) *Surround it again with <div> for each indentation.

```
-----[sample code]-----
<div class="p-2em">
<div class="h-indent-2em">
<p>Man: Peter Piper picked a peck of pickled peppers.</p>
</div>
</div>
```

■Line and Text Spacing

• Line Height

[Notes]

The default setting is 1.75. Consider the line height value when adjusting blank spaces around neighboring lines with margins, etc.

Spaces between lines also change when text sizes change. Unless otherwise instructed, the creator will be responsible for deciding the appropriate spacing.

As a general rule, we do not instruct creators to change spaces or line break text sizes in order to adjust the line spacing.

[CSS class]

class="line-height-normal" : Set line height to the RS default

class="line-height-*em" :Designate line height by the number of characters

Ex: Set the line height to 1.75 times the body

```
-----[sample code]-----
<div class="line-height-3em50">
<p>□Peter Piper picked a peck of pickled peppers.</p>
</div>
```

• Text Spacing

[Notes]

Since spacing may not be displayed between text and image (modern browsers which supports Web Standards do not display it), it is recommended not using in parts which include images and non-standard kanji images.

When designating text spacing in sentences which includes horizontal text in vertical lines, spacing may effect letters which are combined horizontally. Since CSSs do not determine the behavior when text spacing is designated in sentences which includes horizontal text in vertical lines, it is recommended not using in situations such as this.

Designating zero text space at the horizontal text in vertical lines in order to avoid this to happen, be aware now that the spacing between the horizontal letters and the letters below may not function properly.

[CSS class]

class="lspacing-normal" : Set text spacing to the RS default
 class="lspacing-*em" : Designate text space by the number of characters

Ex: Set the text spacing to a quarter the size of a character

```
-----[sample code]-----
<div class="lspacing-0em25">
<p>Text</p>
</div>
```

• Line boundary Character Check Rules

[Notes]

At present, line boundary character check rules differ for each RS, so we cannot regulate them that much.
 Before using these rules, check if the RS you will use supports "-epub-line-break."

[CSS class]

class="line-break-auto" : Set restrictions to the RS default
 class="line-break-loose" : Low restrictions
 class="line-break-normal" : Moderate restrictions
 class="line-break-strict" : High restrictions

Ex: Low restrictions (Does not restrict things like center dots, horizontal ellipses, small script kana.)

```
-----[sample code]-----
<div class="line-break-loose">
<p>Peter • Piper • picked • a • peck • of • pickled • peppers.</p>
<p>"Whoaaaaaaaaa.....h" </p>
</div>
```

• Automatic Line Break Rules

[Notes]

This designates automatic line breaks when text wraps around the end of a line. Word length is not a factor.
 This is used when it becomes problematic to restrict things like the enumeration of western text.

[CSS class]

class="word-break-normal" : Use the standard rules to implement automatic line breaks
 class="word-break-break-all" : Implement automatic line breaks between arbitrary text (Wraparound text at the end of lines even in the middle of long words.)
 class="word-break-keep-all" : Do not automatically break lines in the middle of words (Only wrap around text at spaces or punctuation marks.)

Ex:

```
-----[sample code]-----
<div class="word-break-break-all">
<p>ABCDEFGHJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrstuvwxyz</p>
</div>
```

• Rules for long words

[Notes]

Long strings of western text without spaces or punctuation marks may run off the screen.
 If that happens, use "word-wrap-break-word."
 This guide's default CSS designates { word-wrap: break-word; } in the <body>.

[CSS class]

class="word-wrap-normal" : Standard line break rules

`class="word-wrap-break-word"` : Automatically break lines, wrapping text at the end of a line around to the next line even if in the middle of a word

Ex:

```
-----[sample code]-----
<div class="word-wrap-break-word">
<p>http://abcdefghijklmnopqrstuvwxyz.html</p>
</div>
-----
```

■ Separators

• Separator Lines

[HTML elements]

`<hr/>` : Separator line (horizontal lines in horizontally oriented text, vertical lines in vertically oriented text)

Ex:

```
-----[sample code]-----
<p>Text</p>
<hr/>
<p>Text</p>
-----
```

• Separator Codes (Process as normal characters)

[Notes]

Unless otherwise instructed, use full-width spaces for lowered text.

Ex:

```
-----[sample code]-----
<p>Text</p>
<p>□□□□ * </p>
<p>Text</p>
-----
```

• Separator Images (Process as normal images)

[Notes]

Unless otherwise instructed, designate images with their normal sizes, and use full-width spaces for lowered text.

Ex:

```
-----[sample code]-----
<p>Text</p>
<p>□□□□ </p>
<p>Text</p>
-----
```

■ Links

[Notes]

Ids are basically assigned to the jump destinations.

• Links to the top of other files

[HTML elements]

`Link source`

Ex: Top location of "p-002.xhtml"

```
-----[sample code]-----
<a href="p-002.xhtml">Link to other file</a>
-----
```

• Links to destinations within pages

[HTML elements]

Link source : Link designation side

Jump destination : Jump destination side (or assign to <div>, <p>, <a>, <h1> - <h6>, etc.)

Ex: To [id="link-001"] location

```
-----[sample code]-----
<p><a href="#link-001">Link to content</a></p>
<p id="link-001">Body of content</p> //Jump to the beginning of the paragraph
<p>Body of content<span id="link-001">link destination</span>Body of content</p> //Jump to a part of the paragraph
-----
```

• Links to any place in a file

[HTML elements]

Link source : Link designation side

Jump destination</p> : Jump destination side

(Or designate to <div>, <p>, <a>, <h1>~<h6>, etc.)

Ex: To [id="toc-001"] location in "p-002.xhtml".

```
-----[sample code]-----
<p><a href="p-002.xhtml#toc-001">Link to heading</a></p>
<h1 id="toc-001">Heading</h1>
-----
```

• Mutual link

[HTML elements]

<p>Link1</p> : Link1

(jump to link 2)

<p>Link2</p> : Link2

(jump to link 1)

• Annotations

[Notes]

In consideration of the convenience of tapping with a finger, not only will annotation codes be assigned links, but actual words designated by annotations will be assigned links as well. If it is difficult to tell what the word field is even when looking at the annotation recipient, assign a field that seems appropriate.

[CSS class]

class="noteref" : Text color is blue, default is text with emphatic lines

class="note" : inline-block element, text size default is "smaller"

class="footnote" : no particular designation

class="super" : superscript (refer to the item "superscript, subscript")

*Used as an annotation sign here

[HTML elements]

<p>Item* </p>
: Annotation code side (Any code, including class, can be designated in the annotation code)

<p>* Item □Text</p>
: Annotation side

<div class="footnote" id="note-001">

```
<p><a class="note" href="File name#noteref-001">* Item</a>□Text</p>
<p>□Text</p>
</div>
```

: Example of designating the annotation side with <div>

Ex:

```
-----[sample code]-----
<p><a class="noteref" id="noteref-001" href="p-002.xhtml#note-001">The Spider's Thread<span class="super">*
</span></a></p>
<p><a class="note" id="note-001" href="p-001.xhtml#noteref-001">*The Spider's Thread</a> □ From
Dostoyevsky's "The Brothers Karamazov, Book 7 Chapter 3: A Single Scallion"</p>
-----
```

[Reference Information]

By inserting "epub:type" which stand for annotation, RSs would be able to use that information.

The way RS would use the information is not specified by EPUB, nor can be designated, so confirm the action of the RS in advance, and consult with the publisher when using.

■ Handling Boxes

• Box Types

[CSS class]

```
class="display-none" : Designate elements to { display: none; }
class="display-inline" : Designate elements to { display: inline; }
class="display-inline-block" : Designate elements to { display: inline-block; }
class="display-block" : Designate elements to { display: block; }
```

Ex: Hide part of the text

```
-----[sample code]-----
<p>abcde<span class="display-none">fghij</span>klmno</p>
-----
```

• Exterior Blank Space (Margins)

[CSS class]

```
class="m-"
```

[Value Details]

m-auto / m-0 / m-***per / m-*em : Designate all margins (auto, zero, % designation, character count designation)

m-top-auto / m-top-0 / m-top-***per / m-top-*em : Designate margin at top of screen

m-bottom-auto / m-bottom-0 / m-bottom-***per / m-bottom-*em : Designate margin at bottom of screen

m-right-auto / m-right-0 / m-right-***per / m-right-*em : Designate margin on right side of screen

m-left-auto / m-left-0 / m-left-***per / m-left-*em : Designate margin on left side of screen

m-start-auto / m-start-0 / m-start-***per / m-start-*em : Designate line start margin

m-end-auto / m-end-0 / m-end-***per / m-end-*em : Designate line end margin

m-before-auto / m-before-0 / m-before-***per / m-before-*em : Designate line front margin

m-after-auto / m-after-0 / m-after-***per / m-after-*em : Designate line back margin

Ex: Set all margins to 2 characters

```
-----[sample code]-----
<div class="m-2em">
<p>Text</p>
</div>
-----
```

Ex: Set side margins to 3 lines (when line height is 1.75)

```
-----[sample code]-----
```

```
<div class="m-right-5em25 m-left-5em25">
<p>Text</p>
</div>
```

• Interior Blank Space (Padding)

[CSS class]
class="p-*

[Value Details]

p-auto / p-0 / p-***per / p-*em : Designate all padding (auto, zero, % designation, character count designation)
 p-top-auto / p-top-0 / p-top-***per / p-top-*em : Designate padding at top of screen
 p-bottom-auto / p-bottom-0 / p-bottom-***per / p-bottom-*em : Designate padding at bottom of screen
 p-right-auto / p-right-0 / p-right-***per / p-right-*em : Designate padding on right side of screen
 p-left-auto / p-left-0 / p-left-***per / p-left-*em : Designate padding on left side of screen

p-start-auto / p-start-0 / p-start-***per / p-start-*em : Designate line start padding
 p-end-auto / p-end-0 / p-end-***per / p-end-*em : Designate line end padding
 p-before-auto / p-before-0 / p-before-***per / p-before-*em : Designate line front padding
 p-after-auto / p-after-0 / p-after-***per / p-after-*em : Designate line back padding

Ex: Set all padding to 2 characters

```
-----[sample code]-----
<div class="p-2em">
<p>Text</p>
</div>
```

• Size

[Notes]

When designating sizes with the line length direction and the line width direction rather than with height and width, be aware that you cannot use both fixed values and max values at the same time within the same element. (This is because when changing the orientation direction, you have to clear the other size.) If you are not trying to switch between vertical and horizontal orientations, and are not thinking of combining them, we recommend designating the size with height and width.

*You can simultaneously designate line length directions and line width directions themselves by making them a nest as shown below.

However, it is safest not to restrict the max width direction (page progression direction) value, and not count on any actions just yet.

```
-----[sample code]-----
<div class="measure-10em">
<div class="extent-5em25">
<p>Content</p>
</div>
</div>
```

[CSS class]
 class="height-*"
 class="width-*"
 class="measure-*"
 class="extent-*"
 class="max-height-*"
 class="max-width-*"
 class="max-measure-*"
 class="max-extent-*"
 class="max-size-*

[Value Details]

height-auto / height-***per / height-*em : Designate height

width-auto / width-***per / width-*em : Designate width

measure-auto / measure-***per / measure-*em : Designate line length direction size

extent-auto / extent-***per / extent-*em : Designate line width direction size

max-height-none / max-height-***per / max-height-*em : Set height to maximum value

max-width-none / max-width-***per / max-width-*em : Set width to maximum value

max-measure-none / max-measure-***per / max-measure-*em : Set line length direction size to maximum value

max-extent-none / max-extent-***per / max-extent-*em : Set line width direction size to maximum value

max-size-none / max-size-***per / max-size-*em : Set height and width to same maximum value

Ex: Set size to 10 characters high, 5 lines long (when line height is 1.75)

```
-----[sample code]-----
<div class="height-10em width-8em75">
<p>Text</p>
</div>
```

■ Ruled Lines

• Borders

[Notes]

When enclosing page-fit images with borders, the borders run off the page.

For this reason, this guide prohibits applying ruled lines, including borders, to page-fit images.

[CSS class]

class="k-solid" : Solid line border

class="k-dotted" : Dotted line border

class="k-double" : Double line border

class="k-dashed" : Dashed line border

class="k-solid-black" : Solid line border (black)

class="k-solid-gray" : Solid line border (gray)

class="k-solid-silver" : Solid line border (silver)

class="k-solid-white " : Solid line border (white)

Ex: Enclosing an in-line element with a border

```
-----[sample code]-----
<p>abcde<span class="k-solid">fghij</span>klmno</p>
```

Ex: Enclosing a block element with a border

```
-----[sample code]-----
<div class="k-solid">
<p>Text</p>
</div>
```

• Ruled Lines

[CSS class]

Line Types

class="k-solid-*" : Solid line

class="k-dotted-*" : Dotted line

class="k-double-*" : Double

class="k-dashed-*" : Dashed

[Value Details]

*Top of screen/Bottom/Right/Left/Top-Bottom/Left-Right

k-solid-top / k-solid-bottom / k-solid-right / k-solid-left
 / k-solid-topbottom / k-solid-rightleft
 k-dotted-top / k-dotted-bottom / k-dotted-right / k-dotted-left
 / k-dotted-topbottom / k-dotted-rightleft
 k-double-top / k-double-bottom / k-double-right / k-double-left
 / k-double-topbottom / k-double-rightleft
 k-dashed-top / k-dashed-bottom / k-dashed-right / k-dashed-left
 / k-dashed-topbottom / k-dashed-rightleft

*Line start/Line end/Line front/Line back/Line start-end/Line front-back
 k-solid-start / k-solid-end / k-solid-before / k-solid-after
 / k-solid-startend / k-solid-beforeafter
 k-dotted-start / k-dotted-end / k-dotted-before / k-dotted-after
 / k-dotted-startend / k-dotted-beforeafter
 k-double-start / k-double-end / k-double-before / k-double-after
 / k-double-startend / k-double-beforeafter
 k-dashed-start / k-dashed-end / k-dashed-before / k-dashed-after
 / k-dashed-startend / k-dashed-beforeafter

Line Widths

class="k-*px" : 0-8px
 class="k-thin" : Thin line
 class="k-medium" : Medium line
 class="k-thick" : Thick line

Ex: Applying a left-right line (4px) to a block element

-----[sample code]-----
 <div class="k-solid-rightleft k-4px">
 <p>Text</p>
 </div>

■Aligning Block Elements

[Notes]

The horizontal alignment of vertically oriented elements, and the vertical alignment of horizontally oriented elements, does not function.

[CSS class]

class="block-align-left" : [For horizontal orientations] Left side of screen
 class="block-align-center" : [For horizontal orientations] Horizontal center of screen
 class="block-align-right" : [For horizontal orientations] Right side of screen
 class="block-align-top" : [For vertical orientations] Top of screen
 class="block-align-middle" : [For vertical orientations] Vertical middle of screen
 class="block-align-bottom" : [For vertical orientations] Bottom of screen
 class="block-align-start" : Start of line
 class="block-align-center" : Center of line
 class="block-align-end" : End of line

Ex: Aligning a box 4-characters high enclosed in a border to the end of a line

-----[sample code]-----
 <div class="height-4em k-solid block-align-end">
 <p>Text</p>
 </div>

■Page Breaks in the Same File

[Notes]

As a general rule, page breaks serve to change files out of consideration for such things as the processing capabilities

of devices.

If you want to insert a page break within the same file for an unavoidable reason, such as to prevent the number of files from increasing, use the below example to designate the page break.

Note that this applies only to files like series of short writings and column-like articles that do not need file-wide stylistic changes, and do not have a lot of content on each page.

Note that not many of the RSs currently available can use this, so be sure to carefully check the capabilities of the RS you plan to use.

[CSS class]

class="pagebreak" : Insert page break immediately after the designated block

class="pagebreak-before" : Insert page break immediately before the designated block

class="pagebreak-both" : Insert page break before and after the designated block

Ex: Inserting a page break in the same file for something like a collection of haiku poems, in which there is only **one** line of text on a single page.

-----[sample code]-----

```
<div class="pagebreak">
<p>Peter Piper picked a peck of pickled peppers.</p>
</div>
<div class="pagebreak">
<p>Hickory Dickory Dock</p>
</div>
```

■ Designating Colors

[Notes]

If you need different colored classes that can do the same things, they can be added to the CSS.

Refer to the section entitled "Default CSS Files" later in this guide to learn more.

Keep in mind that that areas designated with bright yellow colors and the like may not be visible in monochrome devices, etc.

• Text Colors

[CSS class]

*Text colors for 1C

```
class="color-black" : { color: #000000; }
class="color-dimgray" : { color: #696969; }
class="color-gray" : { color: #808080; }
class="color-darkgray" : { color: #a9a9a9; }
class="color-silver" : { color: #c0c0c0; }
class="color-gainsboro" : { color: #dcdcdc; }
class="color-white" : { color: #ffffff; }
class="color-transparent" : { color: transparent; }
```

*Basic Colors

```
class="color-red" : { color: #ff0000; }
class="color-blue" : { color: #0000ff; }
class="color-cyan" : { color: #00ffff; }
class="color-magenta" : { color: #ff00ff; }
class="color-orangered" : { color: #ff4500; }
```

Ex:

-----[sample code]-----

```
<p><span class="color-silver">Text</span></p>

<div class="color-silver">
<p>Text</p>
<p>Text</p>
</div>
```

• Background Colors

[CSS class]

*Background colors for 1C

```
class="bg-black" : { background-color: #000000; }
class="bg-dimgray" : { background-color: #696969; }
class="bg-gray" : { background-color: #808080; }
class="bg-darkgray" : { background-color: #a9a9a9; }
class="bg-silver" : { background-color: #c0c0c0; }
class="bg-gainsboro" : { background-color: #dcdcdc; }
class="bg-white" : { background-color: #ffffff; }
class="bg-transparent" : { background-color: transparent; }
```

*Basic Colors

```
class="bg-red" : { background-color: #ff0000; }
class="bg-blue" : { background-color: #0000ff; }
class="bg-cyan" : { background-color: #00ffff; }
class="bg-magenta" : { background-color: #ff00ff; }
class="bg-orangered" : { background-color: #ff4500; }
```

Ex:

```
-----[sample code]-----
<p><span class="bg-silver">Text</span></p>

<div class="bg-silver">
<p>Text</p>
<p>Text</p>
</div>
-----
```

• Inverted Text Colors (Black and White Inversion)

[CSS class]

class="inverse"

Ex:

```
-----[sample code]-----
<p><span class="inverse">Text</span></p>

<div class="inverse">
<p>Text</p>
<p>Text</p>
</div>
-----
```

• Ruled Line Colors

[CSS class]

*Ruled line colors for 1C

```
class="k-black" : { border-color: #000000; }
class="k-dimgray" : { border-color: #696969; }
class="k-gray" : { border-color: #808080; }
class="k-darkgray" : { border-color: #a9a9a9; }
class="k-silver" : { border-color: #c0c0c0; }
class="k-gainsboro" : { border-color: #dcdcdc; }
class="k-white" : { border-color: #ffffff; }
class="k-transparent" : { border-color: transparent; }
```

*Basic Colors

```
class="k-red" : { border-color: #ff0000; }
class="k-blue" : { border-color: #0000ff; }
```

```
class="k-cyan" : { border-color: #00ffff; }
class="k-magenta" : { border-color: #ff00ff; }
class="k-orangered" : { border-color: #ff4500; }
```

Ex: Red border

```
-----[sample code]-----
<p><span class="k-solid k-red">Text</span></p>

<div class="k-solid k-red">
<p>Text</p>
<p>Text</p>
</div>
-----
```

• Link Colors

[Notes]

These can be customized by editing the CSS files.

Refer to the section entitled "Default CSS Files" later in this guide to learn more.

[Reference Information] *Not recommended by this guide.

■ Wraparound

[Notes]

Displays may be greatly distorted when using this with clear, such as WebKit products, so be sure to carefully check the capabilities of the target RS when using it.

[CSS class]

```
class="float-none" : No wraparound
class="float-start" : Wrap around in the line start direction
class="float-end" : Wrap around in the line end direction
class="float-clear" : Deactivate wraparound (clear: both; )
class="float-clear-start" : Deactivate wraparound in the line start direction
class="float-clear-end" : Deactivate wraparound in the line end direction
```

Ex: Wrap text around to an image's line end side (A single-character space separates the image from the text.)

```
-----[sample code]-----
<p></p>
<p>Text</p>
-----
```

Ex: Wrap text around to a block element's line end side (A single-character space separates the element from the text.)

```
-----[sample code]-----
<div class="float-start m-end-1em">
<p></p>
<p>Caption</p>
</div>
<p>Text</p>
-----
```

Default CSS Files

■ Style Sheet Compositions

[Reflowable]

book-style.css File called from XHTML

Prepare readings for each CSS file, RS specifications and bug countermeasures,
and fields for simple customizations of each work

/* Read the below CSS files with @import */

style-reset.css Style set for resetting

style-standard.css Standard style set

style-advance.css Style set to handle things like the combination of logical directions and orientation
directions (can be excluded)

(style-***.css) Style set for individual publishers (Prepare for each company if necessary. Self-
responsible for displays.)

(style-check.css) Style to erect the font in vertical orientations and check the screen in Windows browser
environments (Deactivate reading at time of delivery.)

[Fixed Layouts]

fixed-layout-jp.css File called from XHTML

Do not use other styles so that @import does not need to be supported in fixed layouts.

■ CSS File Operation Rules

CSS reading with @import

This time, in order to unify the XHTML file descriptions, we will only read the main CSS from the XHTML file, and read the other CSS files from within the main CSS with @import.

However, this does not apply if you want to change the CSS files to read based on pages.

*Such as in cases where you want to read individual publisher style sets only on certain pages.

In principle, the default CSS cannot be altered.

As a general rule, the CSS file prepared in advance cannot be altered.

If you need to change any values, use the customization field at the end of "book-style.css", or prepare an individual publisher style set and overwrite it.

Self-responsibility for alterations and additions

This guide does not take into consideration layouts that require complicated designations. If you change CSS contents as below, you will assume responsibility for the content displayed.

Change values set in classes

Add properties to classes

Move description locations to change priority orders.

Change the name of classes linked with other classes.

Add new classes.

Add individual publisher style sheets.

The individual CSS file reading order can be freely switched with others.

Doing things like lightening the amount of files by deleting unnecessary classes in works, or deleting all the comment

lines to facilitate automated tasks should not impair any actions. Even so, be careful not to delete anything essential.

If necessary, you can prepare as many individual CSS files as you like.

However, if the task of reading files all at once becomes too heavy, it may negatively affect performance. Be sure to carefully check things like the capabilities of the target RS in advance.

Avoid duplicate ids

Although in theory it is sufficient for each id to be unique to each page (XHTML file), we have decided to require ids to be unique values throughout an entire work out of consideration for the composition of EPUB data, which is comprised of multiple files.

■ Creating and adding standard classes

[Outline]

The registered items may not provide enough values for some styles such as lowering text and designating sizes. Since it is impossible to register sufficient values for everything in advance, the default style sheet set only contains a few values that seem easy to use.

For example, the font size designations listed only go up to a maximum of 300%. If you need a size that has not been registered in advance, such as 400%, you must register the style yourself.

If you only need to register a few styles, use "Customization Fields by Work" in book-style.css.

In the case of standard classes, we recommend you use the same format as existing classes and only change the values when registering them.

Try to make it easily discernible to others what the class is for. If registering numerous classes, or registering classes you would like to use in the same way on multiple works, prepare individual CSS files and read them in book-style.css with @import.

For instance, if you have special classes for standardly used pages such as imprints and identical pages in a series, they would probably be easier to manage by preparing separate CSS files.

[Standard Class Types]

A. Items whose contents do not require changes when switching between vertically and horizontally oriented pages.

For example, font sizes and color settings use the same designations in both vertical and horizontal orientations. The below is an actual font size setting registered in style-standard.css.

```
-----[sample code]-----
.font-1em20 { font-size: 1.20em; }
-----
```

If you need an unregistered font size such as 1.25em, create a new file changing only the class name and value's size number part as follows:

```
-----[sample code]-----
/* -----
 * Customization Fields by Work
 * ----- */
.font-1em25 { font-size: 1.25em; }
-----
```

Up to two digits can be used in the value after the decimal point, as shown above.

Note that class names cannot use dot symbols, so integers are listed before the unit, and fractional values after.

If there is no fractional value, the class name shall be labeled only with the integer value as follows:

```
-----[sample code]-----
.font-1em { font-size: 1.00em; }
-----
```

If labeling classes with percentages, list them with three digits as follows:

-----[sample code]-----
 .font-085per { font-size: 85%; }

B. Items that require content changes when switching between vertical and horizontal orientations

When designating hanging indents, padding is used from the top of vertical orientated pages and from the left of horizontally oriented pages to always indent the content from the start of the line with the same class name.

The following is a hanging indent actually registered in style-standard.css.

-----[sample code]-----
 .hltr .h-indent-1em { text-indent: -1em; padding-left: 1em; }
 .vrtl .h-indent-1em { text-indent: -1em; padding-top: 1em; }

Add "hltr" and a half-width space before the class you want to apply during horizontal orientations, and ".vrtl" and a half-width space during vertical orientations.

[Reference Information] *Not recommended by this guide.

C. Page orientation nesting measures for items with content that changes when switching between vertical and horizontal orientations.

When vertical and horizontal orientations are combined, the display collapses with only the B settings.

For example, when you create a vertical block in a horizontally-oriented page, and you indent the hanging text in it as described above, both the horizontal padding from the left and the vertical padding from the top will be simultaneously applied.

To counter this, you must delete the horizontal padding from the left.

Below is the sample code for setting the padding from the left to zero when vertically oriented text protrudes on a horizontally oriented page, and for setting the padding from the top to zero when horizontally oriented text protrudes on a vertically oriented page.

-----[sample code]-----
 .hltr .vrtl .h-indent-1em { padding-left: 0; }
 .vrtl .hltr .h-indent-1em { padding-top: 0; }

Note, however, that this method also has a shortcoming. It occurs when you want to set the padding from the left in addition to setting the indentation.

This kind of case is difficult to handle, so in actual use it is safest to avoid designating both the indentation and the other padding simultaneously in the same element as shown below.

-----[sample code]-----
 <div class="p-top-2em">
 <div class="h-indent-1em">
 <p>Text</p>
 </div>
 </div>

At first appearance descriptions like the following may appear to not require nesting measures, but in actuality, displays may still collapse when horizontal text appears on vertically oriented pages even though no problems occur when vertical text appears on horizontally oriented pages.

-----[sample code]-----
 .hltr .h-indent-1em { text-indent: -1em; padding: 0 0 0 1em; }

```
.vrtl .h-indent-1em { text-indent: -1em; padding: 1em 0 0 0; }
```

Contrarily, when switching the listed order of top-to-bottom lines, the display collapses when vertical text appears on horizontally oriented pages.

This is due to the CSS rule of prioritizing more recently read styles.

For the same reason, even this guide's default CSSs cause the screen to collapse when there are two orientation direction nests.

Carefully consider the consequences when deciding whether or not to combine orientation directions.

In addition to hanging indents, similar types of problems due to combining orientation directions occur in classes with displays that change based on page orientations, such as lowering and raising text, margins and padding, logical direction classes such as ruled lines, emphatic lines, underlined links, and image baselines.

■Customizations

Several classes that can be overwritten and blank classes have been prepared in advance in .book-style.css.

For example, if you want to delete a link line, you can use a class that can be overwritten as follows:

```
-----[sample code]-----
/* Basic setting (Top: Horizontally oriented; Bottom: Vertically oriented) */
.hltr a {
}
.vrtl a {
}
```

↓

```
-----[sample code]-----
/* Basic setting (Top: Horizontally oriented; Bottom: Vertically oriented) */
.hltr a {
text-decoration: none;
}
.vrtl a {
text-decoration: none;
}
```

*Add descriptions if countermeasures are needed for combining orientation directions.

When using a blank class, you are responsible for carefully considering which RS to use.

If you want to make the display of your publication safer on as many RSs as possible, we recommend you do not use any CSS properties other than the ones assumed in this guide, and you do not make designations that are too complex.

Note that even if you designate other fonts and text sizes with XHTML in HTML elements that have fonts and text sizes designated in the book-style.css customization field such as the one below, and in elements with designated classes, your designations may not be reflected.

```
-----[sample code]-----
.hltr .oo-midashi {
font-size: 1.5em;
}
.vrtl .oo-midashi {
font-size: 1.5em;
}
```

```
<h1 class="oo-midashi font-100per">Heading</h1> // Is shown at 150% rather than 100%
```

This is based on the CSS priority order. If you encounter difficulties with it, create a new class, etc.

■ Creating your own CSS files

As a basic rule, we recommend you save files in UTF-8 (no BOM), and always insert `[@charset "UTF-8";]` at the start of all files.