

# - 5 資料集 -



## 5. 資料集

### 5.1 入出力プロパティ

以下は、各種プログラミング言語からCSSJにアクセスする際に設定できるプロパティの一覧です。プロパティの設定方法の詳細は[開発マニュアル](#)をご参照ください。

表 5.1 入力関連プロパティ

名前	デフォルト	バージョン	説明
input.default-encoding	JISAutoDetect	CSSJ 1.0.0	HTMLファイルのデフォルトのキャラクタ・エンコーディング名です。CSSJはMETA要素をもとにHTMLのキャラクタ・エンコーディングを判断しますが、それで判断できない場合はこのプロパティを使います。
input.filters	x s l t default-to-xhtml loose-html	CSSJ 1.0.0	入力HTML(XML)のフィルタリングです。適用される順にフィルタ名をスペース区切りで並べます。以下のフィルタ名が用意されています。 <b>xslt</b> <xml-stylesheet...処理命令で指定されているXSL変換スタイルシートを適用します。 <b>default-to-xhtml</b> XMLで名前空間が指定されていない要素をXHTML名前空間に変換します。 <b>loose-html</b> 一般的なHTMLを解釈できるようにするためのフィルタです。
input.default-stylesheet	N/A	CSSJ 1.0.0	デフォルトのCSSスタイルシートのURIです。このプロパティが指定されている場合、最初にデフォルトのスタイルシートが読み込まれます。
input.stylesheet.titles	N/A	CSSJ 1.0.0	適用するCSSスタイルシートのタイトルをスペース区切りで並べます。link要素またはxml-stylesheet処理命令で関連付けられたスタイルシートについて、デフォルトでは代替スタイル以外が全て適用されますが、このプロパティを用いて適用するスタイルシートを指定することができます。
input.xslt.default-stylesheet	N/A	CSSJ 1.2.0	デフォルトのXSLTスタイルシートのURIです。このプロパティが指定されている場合、最初にデフォルトのスタイルシートが読み込まれます。input.filtersにxsltフィルタが存在するときのみ有効です。

表 5.2 HTTPアクセス関連プロパティ

名前	デフォルト	バージョン	説明
input.http.referer	true	CSSJ 1.0.1	HTTP通信でデータを取得するときにRefererヘッダを送るかどうかの指定です。trueまたはfalseで指定します。trueを指定すると、HTMLファイルのURLをRefererヘッダとして送ります。falseを指定すると、Refererを用いて画像などのリソースへの直接アクセスを規制しているサイトでリソースにアクセスできなくなります。
input.http.proxy.host	N/A	CSSJ 1.2.6	プロキシのホスト名です。この設定するとHTTP通信でプロキシをします。
input.http.proxy.port	8080	CSSJ 1.2.6	プロキシを使う際のポート番号です。この設定はinput.http.proxy.hostが設定されている場合のみ有効です。
input.http.authentication.preemptive	false	CSSJ 1.2.6	HTTP通信で認証を行う場合に、最初から認証情報を送るかどうかの設定です。trueまたはfalseで指定します。trueを指定すると、Authorizationヘッダ等の認証情報を最初の接続で送ります。falseを指定すると、最初にサーバーから401レスポンスを受け取ってレルムや認証スキーマ等の情報を取得します。trueを指定した場合、Digest認証や複数のレルムが存在するサーバーで認証が行われなくなります。
input.http.proxy.authentication.user input.http.proxy.authentication.password	N/A	CSSJ 1.2.6	認証が必要なプロキシサーバーでの認証情報(user,password)です。この設定はinput.http.proxy.hostが設定されている場合のみ有効です。
input.http.authentication.n.host input.http.authentication.n.user input.http.authentication.n.password input.http.authentication.n.port input.http.authentication.n.realm input.http.authentication.n.schema	N/A	CSSJ 1.2.6	認証が必要なサイトでの認証情報です。nは通し番号で、nが同じ4つのプロパティで一組です。hostは対象となるホストで、user,passwordが認証に用いられるユーザー名とパスワードです。portはポート番号、realmは認証領域のレルム、schemaは認証スキーマ(basicまたはdigest)です。port,realm,schemaは省略可能で、省略した場合はそれぞれ全てのポート、レルム、スキーマでの認証が行われます。通し番号は0から開始してカウントしていき、必要な情報(hostおよびuser)が欠けていた時点で以降のパラメータは無効となります。
input.http.cookie.n.domain input.http.cookie.n.name input.http.cookie.n.value input.http.cookie.n.path	N/A	CSSJ 1.2.6	送信するクッキーです。nは通し番号で、nが同じ4つのプロパティで一組です。domain,name,valueはそれぞれクッキーのドメインと名前、値です。pathはクッキーのパスで、省略した場合はルート(/)となります。通し番号は0から開始してカウントしていき、必要な情報(domainおよびname)が欠けていた時点で以降のパラメータは無効となります。

表 5.3 出力関連プロパティ

名前	デフォルト	バージョン	説明
output.type	application/pdf	CSSJ 1.0.0	出力(MIME)形式です。以下の値が利用できます。 <b>application/pdf</b> PDFファイル
output.page-width	210mm	CSSJ 1.0.0	ページの幅です。デフォルトはA4の横幅です。CSSの長さの単位(mm,cm,in,pt,pc,px)を使ってください。
output.page-height	297mm	CSSJ 1.0.0	ページの高さです。デフォルトはA4の高さです。CSSの長さの単位(mm,cm,in,pt,pc,px)を使ってください。
output.auto-height	false	CSSJ 1.0.0	自動高さの指定です。falseまたはtrueで指定します。trueにすると、自動改ページをせず、ページの高さを文書の内容の高さに合わせます。このとき、output.page-heightプロパティは無効になります。
output.marks	none	CSSJ 1.0.0 / hidden は CSSJ 1.2.1	トンボおよび裁ち口の表示です。none,crop,cross,both,hiddenのいずれかを指定します。それぞれ、トンボ・裁ち口なし、コーナートンボを表示、センタートンボを表示、両方のトンボを表示、裁ち口だけを表示する、という意味になります。
output.size-limit	N/A	CSSJ 1.2.0	出力データの最大サイズ(バイト)です。サイズが限界に達すると、処理が中断されます。デフォルトでは無制限です。
output.page-limit	N/A	CSSJ 1.2.0	最大ページ数です。ページ数が限界に達すると、処理が中断されます。デフォルトでは無制限です。
output.color	rgb	CSSJ 1.2.1	出力結果のカラー・タイプです。rgb,grayで指定します。rgbではフルカラーで出力されます。grayでは、グレースケールに変換されます。
output.broken-image	cross	CSSJ 1.2.2	壊れた画像の表示方法。hidden,crossで指定します。hiddenでは空白となります。crossでは、×印が表示されます。

表 5.4 Java2D(画面・画像・印刷)出力関連プロパティ

名前	デフォルト	バージョン	説明
output.j2d.antialiasing	false	CSSJ 1.0.0	アンチエイリアスの指定です。falseまたはtrueで指定します。trueにすると、アンチエイリアス(ジャギー消し)をします。
output.j2d.pdf-layout	true	CSSJ 1.0.0	PDFレイアウトの再現の指定です。falseまたはtrueで指定します。trueにすると、画面表示、画像の出力がPDFと全く同じレイアウトになるため、PDFのプレビューとして利用することができます。ただし、フォントの違いから不自然になる可能性があります。

表 5.5 PDF出力関連プロパティ

名前	デフォルト	バージョン	説明
output.pdf.platform-encoding	MS932	CSSJ 1.2.0	PDFを表示する環境のプラットフォームのキャラクタ・エンコーディングです。 PDF1.2以前ではフォント名が影響を受けます。PDF1.3以降ではUTF-8が使われるため無関係です。 すべてのバージョンのPDFにおいて添付ファイル名が影響を受けます。ファイル名にマルチバイト文字が使われている場合、このエンコーディングが表示するプラットフォームのものとは一致しないと文字化けします。日本語の文書であればMS932(Windows版Shift_JIS)、韓国語であればEUC-KR、繁体字中国語ではBig5といった指定をしてください。
output.pdf.hyperlinks	false	CSSJ 1.0.0 / PDF 1.2	ハイパーリンク機能です。falseまたはtrueで指定します。 trueにすると、PDFからWWWなどへのハイパーリンクが有効になります。
output.pdf.hyperlinks.href	relative	CSSJ 1.1.0 / PDF 1.2	ハイパーリンクのアドレスの記述方法です。relativeまたはabsoluteで指定します。 relativeでは相対アドレス指定となり、HTMLのa要素のhref属性がそのまま使われます。absoluteでは絶対URIに変換されてPDFに反映されます。
output.pdf.bookmarks	false	CSSJ 1.0.0 / PDF 1.2	ブックマーク機能です。falseまたはtrueで指定します。 trueにすると、H1～H6要素をもとにブックマーク(アウトライン)を生成します。
output.pdf.compression	binary	CSSJ 1.0.0 / PDF 1.2	圧縮方法です。none,ascii,binaryで指定します。 後者ほど圧縮効率がよくなります。noneでは画像以外は圧縮せず、asciiでは画像以外の内容も圧縮されますが、生成されるPDFはテキストファイルとなります。binaryの場合、生成されるPDFは圧縮され、かつバイナリ形式となります。 ただし、暗号化を行う場合は、結果的に全てバイナリとなることに注意してください。

表 5.5 PDF出力関連プロパティ

名前	デフォルト	バージョン	説明
output.pdf.version	1.4	CSSJ 1.1.0	出力されるPDFファイルのバージョンです。1.2,1.3または1.4が指定可能です。PDFのバージョンによって利用できる機能が変わります。指定されたバージョンで未対応の機能を使おうとすると警告が出力され、PDFに反映されません。
output.pdf.fonts.policy	generic	CSSJ 1.1.0 / PDF 1.2	フォントの使用に関するポリシーです。generic,external,embedで指定します。genericでは基本14フォントと、一般フォント以外用いません。externalではその他のフォントも用いますが、埋め込みはしません。embedでは全てのフォントを用いて、かつ可能であればフォントの埋め込みを行います。
output.pdf.jpeg-image	raw	CSSJ 1.1.0 / PDF 1.2	JPEG画像の埋め込み方法です。raw,to-flateで指定します。rawでは元のデータをそのまま使います。to-flateではFlateDecode形式に変換されます。
output.pdf.encryption	none	CSSJ 1.2.0 / PDF 1.2 ただしv2暗号化はPDF 1.3	暗号化方式です。none,v1,v2で指定します。noneは暗号化なし、v1は40ビットArcfour暗号、v2は40-128ビットArcfour暗号です。
output.pdf.encryption.length	説明を参照	CSSJ 1.2.0 / PDF 1.3	暗号化キーの長さ(ビット)です。output.pdf.encryption=v1では40で固定です。v2では40から128の間で、8ビット刻みで指定可能です。
output.pdf.encryption.user-password	空	CSSJ 1.2.0 / PDF 1.2	文書を開くためのパスワードです。このパスワードを使って文書を開く場合は、文書に設定されたパーミッションによる制限がかかります。
output.pdf.encryption.owner-password	ユーザーのパスワード	CSSJ 1.2.0 / PDF 1.2	文書の権限を変更するためのパスワード(マスタパスワード)です。文書に対するあらゆる操作を可能にします。
output.pdf.encryption.permissions.print	true	CSSJ 1.2.0 / PDF 1.2	文書を印刷する権限です。true=許可,false=禁止です。
output.pdf.encryption.permissions.modify	true	CSSJ 1.2.0 / PDF 1.2	文書中の内容を変更をする権限です。true=許可,false=禁止です。
output.pdf.encryption.permissions.copy	true	CSSJ 1.2.0 / PDF 1.2	文書中のテキストや画像をコピーする権限です。true=許可,false=禁止です。

表 5.5 PDF出力関連プロパティ

名前	デフォルト	バージョン	説明
output.pdf.encryption.permissions.add	true	CSSJ 1.2.0 / PDF 1.2	注釈を追加・変更する、あるいはフォームに入力する権限です。output.pdf.encryption.permissions.modify=trueであればフォームの追加・変更も許可されます。true=許可,false=禁止です。
output.pdf.encryption.permissions.fill	true	CSSJ 1.2.0 / PDF 1.3、ただしv2暗号化のみ	フォームに入力する権限です。true=許可,false=禁止です。
output.pdf.encryption.permissions.extract	true	CSSJ 1.2.0 / PDF 1.3、ただしv2暗号化のみ	障害のあるユーザーのために文書中のテキストや画像を抽出する権限です。true=許可,false=禁止です。
output.pdf.encryption.permissions.assemble	true	CSSJ 1.2.0 / PDF 1.3、ただしv2暗号化のみ	文書中に新しいページ、ブックマーク、サムネイル画像を追加する権限です。true=許可,false=禁止です。
output.pdf.encryption.permissions.print-high	true	CSSJ 1.2.0 / PDF 1.3、ただしv2暗号化のみ	文書を高画質で印刷する権限です。true=許可,false=禁止です。
output.pdf.attachments. <i>n</i> .name output.pdf.attachments. <i>n</i> .description output.pdf.attachments. <i>n</i> .mime-type output.pdf.attachments. <i>n</i> .uri	なし	CSSJ 1.2.0 / PDF 1.4	添付ファイルです。 <i>n</i> は通し番号で、 <i>n</i> が同じ4つのプロパティで一組です。nameはファイル名、descriptionはファイルについての説明で、省略可能です。mime-typeはファイルのMIME型で、省略可能です。uriはファイルの内容が置かれたURIで、省略できません。通し番号は0から開始してカウントしていき、必要な情報(uri)が欠けていた時点でファイルの添付を終わります。nameにはASCII文字だけを使うことを推奨します(マルチバイト文字を含むことはできませんが、表示環境によっては文字化けします)。マルチバイト文字が含まれる場合は、URLエンコードなどでASCIIに変換したものをnameに使い、実際のファイル名をdescriptionにセットしてください。

表 5.6 その他のプロパティ

名前	デフォルト	バージョン	説明
processing.pass-count	1	CSSJ 1.2.0	1回のフォーマット処理のために、文書を処理する回数です。詳細は <a href="#">2パス以上の変換処理</a> を参照してください。

表 5.6 その他のプロパティ

名前	デフォルト	バージョン	説明
processing.make-toc-before	1	CSSJ 1.2.2	目次の生成のためにページ番号を収集するタイミングです。残りパス数がこの数より多い時にページ番号を収集します。デフォルトは1で、すなわち最後以外のパスで目次に表示されるページ番号を収集します。
processing.repeat-table-caption	true	CSSJ 1.2.4	テーブルのキャプション(display: table-caption;)をページ毎に繰り返すかどうかの指定です。trueの場合はキャプションを繰り返しますが、falseの場合は繰り返しません。

## 5.2 情報

infoレベルのメッセージに出力される情報の一覧です。

名前	値の形式	説明
page-number	数値	これから生成されるページの番号です。最終的なページ番号が総ページ数となります。
heading-title	文字列	見出し(h1 ~ h6)として認識された文字列です。
broken-image-uri	文字列	表示できない画像のURIです。
pass-count	数値	残りパス数です。
annot	文字列	cssj:annot属性で任意の要素に指定された注釈です。

## 5.3 CSSのサポート

### 5.3.1 主な制限事項

#### first-line擬似要素

first-line擬似要素に含まれるインライン要素にはスタイルが適用されません。

### 5.3.2 改ページ可能な場所

次の場所では改ページすることができません。

#### 絶対位置・固定位置で配置されたボックスの内部

これはCSS 2.1の仕様に従ったものです。これらのボックスがページをはみ出す場合の扱いについてはCSSでは規定されていません。

CSSJは、ページをはみ出したボックスはページの端(トンボがある場合は断ち切り線の端)の部分で切り落とします。従って、ページからはみ出た部分の内容は印刷されません。

#### 浮動体の内部

CSS 2.1の仕様では浮動体内部での改ページは出来るだけ避けることが推奨されていますが、CSSJは浮動体内部での改ページを全く行いません。

浮動体がページの下端をはみ出す場合、その浮動体の直前で改ページされます。また、ページの大きさを超える浮動体は、絶対位置で配置されたボックス同様に、ページの端で切り落とされます。また、切り落としを防ぐために、浮動体の大きさはページに対してなるべく小さくしてください。

## テーブルのセルおよびキャプション・ヘッダ・フッタの内部

CSS 2.1の仕様ではテーブル内部での改ページは出来るだけ避けることが推奨されていますが、CSSJはテーブルの行間で改ページを行い、かつセルの内部で改ページすることはありません。

従って、rowspan属性によって連結されたセルが存在しない行間のみで改ページが可能です。行がページの下端をはみ出す場合は、その行の直前の行間で改ページされます。また、ページの大きさを超えるセルは、絶対位置で配置されたボックス同様に、ページの端で切り落とされます。切り落としを防ぐためには、セルの大きさをページに対してなるべく小さくし、rowspan属性によって多数の行を連結することを避けてください。

また、改ページが可能なのはテーブルのボディ(tbody要素内あるいはtbodyタグを省略した場合)内部だけです。キャプション、ヘッダ(thead要素)、フッタ(tfoot要素)内での改ページはできません。レイアウトが崩れたり、切り落とされるのを防ぐために、これらの大きさはページの大きさに対してなるべく小さくしてください。

### 5.3.3 テーブルの改ページ

テーブルにキャプション、ヘッダ、フッタが存在する場合、それらは分割されたテーブルの断片全てに付けられます。また、テーブルのボディには少なくとも1行が存在します。

従って、テーブルの高さはキャプション、ヘッダ、フッタの高さと、ボディの1行の高さを足した高さの合計よりも小さくなることはありません。切り落としを防ぐには、この高さが常にページの高さよりも小さくなるようにしてください。

### 5.3.4 改ページ制御

page-break-before特性またはpage-break-after特性にalways,left,rightのいずれかを指定することで強制改ページできます。強制改ページの指定が有効なのはブロック、テーブル行です。ただし、元々改ページができない場所での強制改ページの指定は無視されます。

また、同じ特性にavoidを指定して前後の改ページを抑制できるのは、絶対配置または浮動体でないブロック、テーブル行です。同様にpage-break-inside特性にavoidを指定することにより内部での改ページを抑制できます。ただし、元々内部での改ページができないボックスに対しては、この指定は意味を持ちません。強制改ページと改ページの抑制が競合する場合、CSS 2.1に従い強制改ページが優先されます。

### 5.3.5 拡張機能

#### @namespace指示子

CSSJは複数の名前空間が混在したXML文書をスタイル付けすることができます。その仕様は、W3Cワーキングドラフト [CSS Namespace Enhancements \(Proposal\)](#) に従っています。現在、Netscape7,Firefox1.0など一部のブラウザがこの機能をサポートしています。ここでは、基本的な使い方のみを紹介しておきます。

CSSスタイルシート中で名前空間の接頭辞(prefix)とURIを指定するためには、@namespace指示子を使って以下のように宣言します。

```
/* デフォルトの名前空間のURIをhttp://www.w3.org/1999/xhtmlとする。 */
@namespace url(http://www.w3.org/1999/xhtml);

/* 接頭辞rdfのURIをhttp://www.w3.org/1999/02/22-rdf-syntax-ns#とする。 */
@namespace rdf url("http://www.w3.org/1999/02/22-rdf-syntax-ns#");
```

#### 例 5.1 名前空間の宣言

スタイルシートの選択子(selector)で接頭辞を使う場合は、'|'で区切ります。(':')でないことに注意してください。)

```
/* 要素のスタイルを指定する。 */
rdf|Description { display: block; }

/* ref:about属性がhttp://foo.com/barであるitem要素のスタイルを指定する。 */
item[rdf|about=http://foo.com/bar] { color: Red; }
```

#### 例 5.2 接頭辞を含む選択子

## ページのヘッダ・フッタ

W3Cの勧告候補である [CSS3 Paged Media Module](#) では、ページの内容の外側にページ番号などを表示するための規格が提案されていますが、CSSJはこれをサポートしません。その代わりに、CSSJ 1.2.0から `-cssj-regeneratable` という、ページ毎に生成される内容のための、拡張プロパティが用意されています。

*position: fixed;*指定されたブロックもページ毎に再生成されますが、CSSJ 1.2.0以降では、この方法を使ってページのヘッダやフッタを生成することは非推奨とします。

`-cssj-regeneratable`の値に `none` 以外を指定したボックス(再生成ボックス)はページ毎に再生成されます。配置方法は、`position: fixed;`指定されたボックスに似ています。ページ毎に再生成されるため、例えば `counter-increment` などのカウンタ操作が指定されている場合、ページ毎にカウンタが操作されることとなります。

再生成ボックスは、上位にあるボックスのスタイルの影響を受けません。すなわち、上位のボックスに `color: Red;` という指定がされている場合も、再生成ボックス内の文字色は黒です。ただし、セレクトは再生成先の上位のボックスの影響を受けます。そのため、再生成ボックスに対してはIDセクタ(#ID形式)または `style` 属性による直接指定を用いてください。

`-cssj-regeneratable`の引数は、その再生成ボックスの識別名です。同じ識別名を持つ再生成ボックスは、同じページに1つしか存在できません。文書中で、同じ名前の再生成ボックスを2度配置すると、2度目の再生成ボックスの配置の際に、1度目の再生成ボックスは再生成されなくなります。

再生成ボックスは、ページ番号を各ページに表示するために有用です。例えば以下のブロックが文書の始めに存在すると、各ページの下部中央にページ番号が表示されます。

```
<style type="text/css">
#page-number {
  -cssj-regeneratable: footer;
  bottom: -1cm;
  text-align: center;
  width: 100%;
}
#page-number:before {
  counter-increment: page;
  content: counter(page);
}
</style>
<div id="page-number"></div>
```

### 例 5.3 ページ番号の生成

また、どのページにブロックが生成されるかによってスタイルを変えるために、本来は `@page` 指示子のためである `:first`, `:left`, `:right` 擬似クラスをセクタで利用することができます。上の例で、左ページでは左寄せで、右ページでは右寄せでページ番号を表示するには、さらに以下の記述をスタイルシートに加えます。

```
:left #page-number {
  text-align: left;
}
:right #page-number {
  text-align: right;
}
```

### 例 5.4 :first,:left,:right擬似クラスの拡張

なお、再生成ボックスが配置される際は、ブロックに対する `top`, `left`, `right`, `bottom`, `width`, `height` プロパティは常にページの範囲を基準とします。

## 5.3.6 各特性のサポート状況

以下の表はW3C CSS2.1仕様の各特性のサポート状況です。

表 5.7 CSSサポート状況一覧

特性	サポート	備考
azimuth	しない	音声スタイルのため、印刷には無関係です。
background-attachment	する	
background-color	する	
background-image	する	
background-position	する	
background-repeat	する	
background	する	
border-collapse	する	
border-color	する	
border-spacing	する	
border-style	する	
border-top	する	
border-right	する	
border-bottom	する	
border-left	する	
border-top-color	する	
border-right-color	する	
border-bottom-color	する	
border-left-color	する	
border-top-style	する	
border-right-style	する	
border-bottom-style	する	
border-left-style	する	
border-top-width	する	
border-right-width	する	
border-bottom-width	する	
border-left-width	する	
border-width	する	
border	する	
bottom	する	
caption-side	する	
clear	する	
clip	する	

表 5.7 CSSサポート状況一覧

特性	サポート	備考
color	する	
content	する	
counter-increment	する	
counter-reset	する	
cue-after	しない	音声スタイルのため、印刷には無関係です。
cue-before	しない	音声スタイルのため、印刷には無関係です。
cue	しない	音声スタイルのため、印刷には無関係です。
cursor	しない	インタラクティブスタイルのため、印刷には無関係です。
direction	しない	左から右へ書く言語(アラビア語・ヘブライ語など)はサポートしていません。
display	する	
elevation	しない	音声スタイルのため、印刷には無関係です。
empty-cells	する	
float	する	
font-family	する	cursive,fantasyフォントは利用できません。
font-size	する	
font-style	する	
font-variant	しない	スモール・キャップフォントは利用できません。
font-weight	する	
font	する	
height	する	
left	する	
letter-spacing	する	
line-height	する	
list-style-image	する	
list-style-position	する	
list-style-type	する	
list-style	する	
margin-right	する	
margin-left	する	
margin-top	する	
margin-bottom	する	
margin	する	

表 5.7 CSSサポート状況一覧

特性	サポート	備考
max-height	する	
max-width	する	
min-height	する	
min-width	する	
orphans	する	
outline-color	しない	インタラクティブスタイルのため、印刷には無関係です。
outline-style	しない	インタラクティブスタイルのため、印刷には無関係です。
outline-width	しない	インタラクティブスタイルのため、印刷には無関係です。
outline	しない	インタラクティブスタイルのため、印刷には無関係です。
overflow	する	
padding-top	する	
padding-right	する	
padding-bottom	する	
padding-left	する	
padding	する	
page-break-after	する	
page-break-before	する	
page-break-inside	する	
pause-after	しない	音声スタイルのため、印刷には無関係です。
pause-before	しない	音声スタイルのため、印刷には無関係です。
pause	しない	音声スタイルのため、印刷には無関係です。
pitch-range	しない	音声スタイルのため、印刷には無関係です。
pitch	しない	音声スタイルのため、印刷には無関係です。
play-during	しない	音声スタイルのため、印刷には無関係です。
position	する	
quotes	する	
richness	しない	音声スタイルのため、印刷には無関係です。
right	する	
speak-header	しない	音声スタイルのため、印刷には無関係です。
speak-numeral	しない	音声スタイルのため、印刷には無関係です。
speak-punctuation	しない	音声スタイルのため、印刷には無関係です。
speak	しない	音声スタイルのため、印刷には無関係です。
speech-rate	しない	音声スタイルのため、印刷には無関係です。

**表 5.7 CSSサポート状況一覧**

特性	サポート	備考
stress	しない	音声スタイルのため、印刷には無関係です。
table-layout	する	
text-align	する	
text-decoration	する	
text-indent	する	
text-transform	する	
top	する	
unicode-bidi	しない	
vertical-align	する	
visibility	する	
voice-family	しない	音声スタイルのため、印刷には無関係です。
volume	しない	音声スタイルのため、印刷には無関係です。
white-space	する	
windows	する	
width	する	
word-spacing	する	
z-index	する	

## 5.4 HTMLのサポート

### 5.4.1 概要

CSSJのCSSフォーマッタはXMLをスタイル付けするものですが、HTMLをスタイル付けされたXMLに変換することにより、HTMLをサポートしています。この機能はinput.filtersプロパティによりデフォルトでオンになっているため、特に意識する必要はありません。

また、META要素、画像、ハイパーリンクなどXMLへの変換には関係なくサポートされる機能もあります。これらの機能はXHTML名前空間(<http://www.w3.org/1999/xhtml>)を用いることで、XML中でも利用可能です。

```
<document xmlns:html="http://www.w3.org/1999/xhtml">
...<html:img html:src="image.jpeg" html:width="100" html:height="200"/>...
</document>
```

#### 例 5.5 XML中でimgタグを使う

XML文書内では大文字小文字が区別されます。XHTMLの様式に従って、タグ名、属性名共に小文字で記述してください。

### 5.4.2 各要素・属性のサポート状況

表 5.8 HTMLサポート状況一覧

要素・機能	属性	サポート	備考
a	href name	する	name属性を用いた文書内リンクとハイパーリンクをサポートします。XML文書中で使用可能です。
abbr		する	
acronym		する	
address		する	
applet	width height hspace vspace alt align	しない	枠だけが表示されます。
area		しない	
b		する	
base	href	する	以降のハイパーリンクなどは、hrefからの相対パスになります。XML文書中で使用可能です。
basefont	size color face	しない	font要素と同じ働きをします。
bgsound		しない	
bdo		しない	
big		する	
blink		しない	
blockquote		する	
body	marginheight marginwidth topmargin leftmargin rightmargin bottommargin bgcolor background bgproperties text link	する	alink,vlink属性はサポートしません。
br	clear	する	
button	disabled	しない	
caption	align valign	する	
center		する	
cite		する	
code		する	
colgroup		する	align,bgcolor,charoff,span,valign,width属性はサポートされません。
col		する	charoff,span属性はサポートされません。
comment		しない	
dd		する	
del		する	
dfn		する	

表 5.8 HTMLサポート状況一覧

要素・機能	属性	サポート	備考
dir	type	する	compact属性はサポートしません。
div	align	する	
dl		する	compact属性はサポートしません。
dt		する	
em		する	
embed	border width height hspace v space alt hidden frameborder units	しない	枠だけ表示されます。
fieldset	align	する	
font	size color face font-weight point-size	する	
form		しない	枠だけ表示されます。
frame		しない	
framset		しない	
h1 h2 h3 h4 h5 h6	align	する	
head		する	
hr	align color noshade size width	する	
html		する	
i		する	
iframe		しない	
ilayer	background bgcolor clip height src visibility width left pagex pagey top z-index	する	above,below属性はサポートしません。
img	src alt border width height hspace v space align	する	XML文書中で使用可能です。ただし、XML文書中ではsrc,width,height属性のみ有効です。alt属性によるツールチップは表示されません。
input	disabled	しない	枠だけ表示されます。
input[type=checkbox]	size	しない	枠だけ表示されます。
input[type=text]	size	しない	枠だけ表示されます。
input[type=password]	size	しない	枠だけ表示されます。
input[type=file]	size	しない	枠だけ表示されます。
input[type=radio]		しない	枠だけ表示されます。
input[type=reset]		しない	枠だけ表示されます。
input[type=button]		しない	枠だけ表示されます。

表 5.8 HTMLサポート状況一覧

要素・機能	属性	サポート	備考
input[type=submit]		しない	枠だけ表示されます。
input[type=image]	src border width height align	しない	img要素と同様に表示されます。
ins		する	
isindex		しない	
kbd		しない	
keygen		しない	
label		しない	
layer	background bgcolor clip height src visibility width left pagex pagey top z-index	する	above,below属性はサポートしません。
legend		しない	ブロックとして表示します。
li	type	する	value属性はサポートしません。
listing		する	
link	rel type media href	する	rel="StyleSheet" type="text/css"の場合にhrefのCSSスタイルシートをリンクします。これはCSSスタイルシートに対するxml-stylesheet処理命令と同様に動作します。link要素でXSLTスタイルシートをリンクすることはできません。
map		しない	
marquee	bgcolor width height hspace vspace	しない	スクロールはしません。
menu	type	する	compact属性はサポートしません。
meta	http-equiv name content	する	XML文書中で使用可能です。HTML文書では、http-equiv="Content-Type" content="text/html; charset=エンコーディング名"という指定を行うことでエンコーディングを指定可能です。また、PDF出力の際の文書情報を設定することができます。詳細は「デザイナーガイド」を参照してください。
multicol		しない	ブロックとして表示されます。
nextid		しない	
nobr		する	
noembed		しない	表示しません。
noframes		する	表示します。
nolayer		しない	表示しません。
noscript		する	表示します。

表 5.8 HTMLサポート状況一覧

要素・機能	属性	サポート	備考
object	border width height hspace vspace alt align(absbottom, absmiddle, texttopを除く)	しない	枠だけ表示されます。
ol	type	する	compact,start属性はサポートしません。
optgroup		しない	
option		しない	
p	align	する	
param		しない	
plaintext		する	
pre	cols width wrap	する	
q		する	
ruby rb rt rp		しない	
s		する	
samp		する	
script		しない	表示しません。
select		しない	枠だけ表示されます。
server		しない	
small		する	
spacer		しない	
span		する	
strike		する	
strong		する	
style	disabled type media	する	XML文書中で使用可能です。
sub		する	
sup		する	
table	width height bgcolor background align hspace vspace border frame rules cellspacing cellpadding	する	bordercolordark, bordercolorlight, cols, summary 属性はサポートされません。
tbody	align bgcolor valign	する	charoff属性はサポートされません。
thead	align bgcolor valign	する	charoff属性はサポートされません。
tfoot	align bgcolor valign	する	charoff属性はサポートされません。
td	bordercolor background bgcolor align valign height width nowrap colspan rowspan	する	charoff, bordercolordark, bordercolorlight属性は サポートされません。 colspan, rowspan属性は "display: table-cell"スタイルが指定されてい る要素に付けることでXML文書でも利用可 能です。

表 5.8 HTMLサポート状況一覧

要素・機能	属性	サポート	備考
th	bordercolor background bgcolor align valign height width nowrap colspan rowspan	する	charoff,bordercolordark,bordercolorlight属性はサポートされません。colspan,rowspan属性は"display: table-cell"スタイルが指定されている要素に付けることでXML文書でも利用可能です。
tr	bordercolor background bgcolor align valign height	する	charoff,bordercolordark,bordercolorlight属性はサポートされません。
textarea	cols rows disabled	しない	枠だけ表示されます。wrap属性は無視されます。
title		する	タイトルバーまたは、PDF文書情報のタイトルとして使われます。
u		する	
ul	type	する	compact属性はサポートされません。
var		する	
wbr		しない	
xmp		する	
dir一般属性		しない	
style一般属性		する	XML文書中で使用可能です。

## 5.5 XML/XSLTのサポート

### 5.5.1 xml-stylesheet処理命令

CSSJがXML文書にXSLTスタイルシート、またはCSSスタイルシートを関連付ける仕様は、W3C勧告の [Associating Style Sheets with XML documents version 1.0](#) に従っています。

```
<?xml-stylesheet href="style.xsl" type="text/xsl"?>
```

#### 例 5.6 XML文書をXSLTスタイルシートで変換する

```
<?xml-stylesheet href="style.css" type="text/css"?>
```

#### 例 5.7 XML文書をCSSスタイルシートでスタイル付けする

デフォルトでは、代用でない全てのprintメディアのスタイルシートが順に適用されます。input.stylesheet.titlesプロパティによって適用するスタイルシートを指定することができます。

### 5.5.2 拡張処理命令

XMLで利用できる、CSSJ独自の処理命令があります。これらは、主にHTMLのheadタグ内に記述する設定をXMLでも可能にするものです。

処理命令は<?名前 値?>という形式で、XML宣言以降の任意の場所に記述することができます。また、処理命令の値は、さらに擬似属性とデータ部に分かれることがあります。擬似属性はHTML/XML要素の属性に似ています。例えば、以下の処理命令はname,mediaという擬似属性を持ち、データ部がスタイルシートになっています。

```
<?jp.cssj.stylesheet type="text/css" media="screen" [
  html {
```

```
background-color: Red;
}
]??
```

例 5.8 擬似属性とデータ部を持つ処理命令

表 5.9 拡張処理命令一覧

名前	バージョン	説明
jp.cssj.stylesheet	CSSJ 1.1.0	これはHTMLのstyle要素の代替機能を提供するものです。type,media属性に対して、同名の擬似属性が用意されています。スタイルシート自体はデータ部に記述します。
jp.cssj.document-info	CSSJ 1.1.0	これはHTMLの<meta name="..." content="...">要素の代替機能を提供するものです。name,content属性に対して、name,value擬似属性が用意されています。
jp.cssj.default-encoding	CSSJ 1.1.0	これはHTMLの<meta http-equiv="Content-Type" content="text/html; charset=...">要素の代替機能を提供するものです。エンコーディング名を値に使用します。
jp.cssj.default-style-type	CSSJ 1.1.0	これはHTMLの<meta name="content-style-type" content="...">要素の代替機能を提供するものです。MIME型を値に使用します。

### 5.5.3 XSLTプロセッサ

CSSJはApache Software Foundationが開発したソフトウェアを使用しています。CSSJサーバーに使用されているXSLTプロセッサはXSLT 1.0,XPath 1.0をサポートしています。無保証のGUI版CSSJはJavaVMのXSLTプロセッサを使用します。

XSLTプロセッサはconfディレクトリ内のjava.propertiesファイル内の javax.xml.transform.TransformerFactoryプロパティによって変更可能です。これによりSaxonのような他のXSLTプロセッサを用いることでパフォーマンスや機能が改善されることがありますが、このような修正については保証の範囲外です。お客様の責任で行ってください。

## 5.6 フォント設定ファイル

CSSJが利用するフォントに関する設定は、resource/media/pdf/fonts.xmlに記述されています。このファイルには、以下の事柄が記述されています。

- cmapファイル
- フォント・ファイル
- システム・フォント
- コア・フォント
- 一般フォント
- デフォルトのフォント
- CSS一般フォント・ファミリ

### 5.6.1 cmapファイル

cmapファイルは、シフトJIS,EUCなどの文字コードと、Adobe社独自のキャラクタ・コードを関連付けるファイルです。これは、後述する一般フォントに関連します。fonts.xmlファイルのcmaps要素に記述されていますが、特にこの設定を変更する必要はありません。

### 5.6.2 フォント・ファイル

フォント・ファイルを追加する場合、fonts.xmlファイルにfont-file要素を追加します。利用可能なフォント・ファイルはTrueTypeフォント(.ttfファイル)です。font-fileのsrc属性に、フォント・ファイルへのパスを記述してください。

font-fileのpolicy属性に"type2"を指定すると、CFF/Type2フォントに変換され、PDFに埋め込まれます。"external"を指定すると、フォントのメトリック情報だけがPDFファイルに記述され、フォント自体は埋め込まれません。"external"を指定した場合、PDFファイルを正確に表示するには、利用するフォントが

インストールされた環境が必要です。フォントの埋め込みをすると、どの環境でもPDFファイルを正確に表示できますが、利用するフォントがライセンス上PDFファイルへの埋め込みが許可されているものである必要があります。CSSJは技術的にフォントが埋め込みライセンスが許可されているものかどうかを判別できないため、利用するフォントのライセンスについては、各フォント・ベンダの使用許諾契約書などをご参照ください。

フォント名はフォント・ファイルから自動的に取得されます。CSSから別の名前でも指定できるようにするためには、font-file要素内にalias要素を追加してください、aliasのname属性にフォントの別名を記述してください。

### 5.6.3 システム・フォント

フォント・ファイルを直接指定する他に、OSにインストールされたシステム・フォント名を指定する方法があります。システム・フォントを利用可能にするには、fonts.xmlファイルにsystem-font要素を追加してください、system-fontのsrc属性にはフォント名を記述します。また、all-system-fonts要素を追加することで、全てのシステム・フォントを利用可能になります。

これらの要素には、フォント・ファイル同様にpolicy属性の設定が可能です。また、system-font要素の内部にはalias要素を含めることができます。

### 5.6.4 コア・フォント

コア・フォントはPDFの標準的な欧文フォントで、AFM(Adobe Font Metrics)ファイルによってメトリック情報が定められています。fonts.xmlのcore-font要素によって設定されます。通常は変更する必要はありません。

### 5.6.5 一般フォント

一般フォントは、フォントの種類のみを指定して、実際にどのようなフォントを利用するかはPDFビューワに任せるものです。メトリック情報として、一般的な等幅和文フォントのものを用いるため、プロポーションアルフォントを指定することは適しません。fonts.xmlのgeneric-font要素によって設定されます。name属性にはフォント名、flags属性にはフォントの種類を示すフラグ、cmap属性にはcmap名前を指定します。通常、この設定を変更する必要はありません。

### 5.6.6 デフォルトのフォント

デフォルトのフォントは、CSSJがCSS内で指定されたフォントを探し、最終的に該当するフォントがなかった場合に使われるフォントです。fonts.xmlファイルのdefault要素のname属性にフォント名を記述してください。このフォント名は、実際に利用可能なフォントを指定してください。

### 5.6.7 CSS一般フォント・ファミリ

CSSではserif,sans-serif,monospace,fantasy,cursiveという5種類の一般フォント・ファミリがあります。fonts.xmlのcss-generic要素で、これらのフォント・ファミリ名と実際のフォントを対応付けることができます。

css-genericのgeneric-family要素には、CSSの一般フォント・ファミリ名を記述してください。font-family要素には実際に使われるフォントのリストをカンマ区切りで記述します。フォントのリストの意味はCSSのfont-familyプロパティと同じで、リストの最初に記述されたフォントが優先的に使われます。

CSSJのデフォルト設定では一般フォントまたはコア・フォントを使うようにしているため、常に外部フォントが使われます。fantasy,cursiveの設定は適当ではありません。フォントの埋め込みを行う場合は、適切に設定しておくことを推奨します。

## 5.7 見出しに関する拡張機能

CSSJでは、見出しから目次を生成する機能、ページのヘッダ・フッタに見出しを表示する機能が追加されています。これらは、T<sub>E</sub>XおよびXSL-FOに対抗する機能として、追加されたものです。これらの機能は非標準で、限定的であり、将来のバージョンでは他の方法に置き換えられる可能性があります。

### 5.7.1 2パス以上の変換処理

CSSJ 1.1.x以前では文書の変換処理は常にワンパスでしたが、CSSJ 1.2.0からは、文書を2度以上処理する機能が加えられています。この目的は、ページ番号によって文書の内容を指し示すためです。HTML/XML中に存在する任意の位置の内容が、変換後のPDFの何ページ目に存在するかは一度変換処

理を行わないと分からないためです。

次に説明する、目次の自動生成を行うためには、[processing.pass-countプロパティ](#)に2以上の値を指定し、複数パスの変換する必要があります。

## 5.7.2 目次の自動生成

CSSJ 1.2.0で目次(Table of Contents)の生成機能が追加されました。目次の生成では、1回目の文書処理で各見出し(h1 ~ h6)が存在するページ番号を収集し、2回目の文書処理で目次を生成します。目次の生成場所は、以下のように独自タグcssj:tocで指定します。

```
<cssj:toc xmlns:cssj="http://www.cssj.jp/ns/cssjml"
  offset="2"/>
```

### 例 5.9 目次の生成

cssj:tocは常に目次に変換されるわけではありません。目次に変換されるのは、以前のパスで見出しの存在するページ番号の収集が行われた場合です。ページ番号の収集は、のこりのパス数が[processing.make-toc-beforeプロパティ](#)より多い場合に行われます。デフォルトは1なので、一番最後のパス以外でページ番号の収集が行われます。

offset属性は目次に含まれるページ番号から引かれる値です。一般的な書籍では前付け(表紙、まえがき、目次)のページ番号をローマ数字で記述し、本文で再び1から始まるアラビア数字のページ番号を別立てで用意することが行われます。しかし、CSSJが目次を生成するためのページのカウンタは、表紙を1ページとした通し番号で行われるため、実際のページ番号に合わせるためには、目次が未生成の状態での前付けのページ数分差し引く必要があります。表紙を1ページとして最後まで通し番号でゆく場合はこの限りではありませんが、目次に使われるページ数も考慮する必要があるため、文書の変換処理が3パス必要になります。

目次と本文の通し番号を別々にしている場合、何らかの理由で3パスの変換処理を実行した場合、意図せず目次も含めた通し番号で目次のページが生成されてしまいます。その場合、processing.make-toc-beforeプロパティを(合計パス数 - 1)に設定することにより、これを防ぐことができます。

生成される目次は、次のようなリストの形式をしています。

```
<ul class="cssj-toc">
  <li><a href="#cssj-heading-1"><span class="cssj-title">1. タイトル1</span><span
class="cssj-page">1</span></a></li>
  <li><a href="#cssj-heading-2"><span class="cssj-title">2. タイトル2</span><span
class="cssj-page">5</span></a></li>
  <li><a href="#cssj-heading-3"><span class="cssj-title">3. タイトル3</span><span
class="cssj-page">8</span></a></li>
  <ul>
    <li><a href="#cssj-heading-4"><span class="cssj-title">3.1. タイトル
4</span><span class="cssj-page">8</span></a></li>
    <li><a href="#cssj-heading-5"><span class="cssj-title">3.2. タイトル
5</span><span class="cssj-page">9</span></a></li>
  </ul>
</ul>
```

### 例 5.10 生成される目次の例

例として、これをT<sub>E</sub>Xライクな見栄えで表示するためには、以下のCSSを使ってください。

```
ul.cssj-toc, ul.cssj-toc ul {
  list-style: none;
}
ul.cssj-toc {
  margin: 0;
}
ul.cssj-toc ul {
  margin: 1em;
}
ul.cssj-toc li {
```

```
margin: 1em 1em 1.5em 0;
font-family: sans-serif;
height: 0.5em;
border-bottom: 1pt dotted;
}
ul.cssj-toc ul li {
margin: 0 0 0.5em 1em;
font-family: serif;
}
ul.cssj-toc span {
background-color: White;
padding: 0 0.5em;
}
ul.cssj-toc span.cssj-page {
position: absolute;
right: 0;
}
```

例 5.11 目次を整形するCSSの例

### 5.7.3 ヘッダ・フッタへの見出しの表示

---

ページのヘッダやフッタに、現在のページに記述されているセクションに対応する見出しを表示することができます。そのための拡張として、contentプロパティで利用できる`-cssj-heading`関数が用意されています。

```
content: -cssj-heading(1) ' - ' counter(page);
```

例 5.12 ヘッダの表示

`-cssj-heading`の引数は、表示する見出しのレベルです。この値はh1～h6の数字部分です。例えば、`-cssj-heading(2)`では直前にh2要素に記述された見出しが表示されます。

### 5.7.4 注釈

---

`cssj:annot`属性を任意のタグに設定することにより、CSSJドライバに、現在処理しているページに含まれる注釈を返すことができます。`cssj:annot`の値は、`annot`という名前の情報でドライバに返されます。アプリケーションが、現在CSSJがドキュメントのどの部分进行处理しているのかを判断するために利用できます。