

# ウェブ技術によるルビの事例

2020年3月20日

Author:

[Florian Rivoal](#) (Rivoal Technology)

## 1. CSSとHTMLによるルビの出来るバリエーションの例

### 1.1. 同じマークアップでルビの表示の仕方を切り替える

	<pre>&lt;ruby&gt;   &lt;rb&gt;振&lt;rb&gt;り&lt;rb&gt;仮&lt;rb&gt;名   &lt;rp&gt;(   &lt;rt&gt;ふ&lt;rt&gt;り&lt;rt&gt;が&lt;rt&gt;な   &lt;rp&gt; &lt;/ruby&gt;</pre>	<pre>&lt;ruby&gt;   &lt;rb&gt;今日   &lt;rp&gt;(   &lt;rt&gt;きょう   &lt;rp&gt; &lt;/ruby&gt;</pre>	<pre>&lt;ruby&gt;   &lt;rb&gt;京&lt;rb&gt;都   &lt;rp&gt;(   &lt;rt&gt;きょう&lt;rt&gt;と   &lt;rp&gt; &lt;/ruby&gt;</pre>
初期値のCSS	ふ が な 振り仮名	きょう 今日	きょうと 京都
<pre>rt { display: none; }</pre>	振り仮名	今日	京都
<pre>ruby { display: inline; } rb { display: none; } rt {   display: inline;   font-size: inherit;   font-variant: initial; }</pre>	ふりがな	きょう	きょうと
<pre>rt, ruby, rb, rp {   display: inline;   font-size: unset;   font-variant: initial; }</pre>	振り仮名(ふりがな)	今日(きょう)	京都(きょうと)

※ 現在のcss-rubyレベル1によって、同じマークアップで混ぜルビ（例：「振りかな」もしくは「<sup>ふ</sup>振りがな」）は対応がまだ出来ません。

## 1.2. 同じマークアップでルビの表示の仕方を学年に合わせて切り替える

学年	CSS	表示結果
		<pre>&lt;ruby&gt;   &lt;rb&gt;昆&lt;rb&gt;虫   &lt;rt&gt;こん&lt;rt class=l3&gt;ちゅう &lt;/ruby&gt;</pre>
一年生	<pre>ruby { display: inline; } rb { display: none; } rt {   display: inline;   font-size: inherit;   font-variant: initial; }</pre>	こんちゅう
二年生	初期値	<small>こんちゅう</small> 昆虫
三年生	<code>rt.l3 { visibility: hidden; }</code>	<small>こん</small> 昆虫
四年生	<code>rt { display: none; }</code>	昆虫

## 1.3. 両側ルビの対応

	マークアップ	表示結果
	<pre>&lt;ruby&gt;   &lt;rb&gt;東&lt;rb&gt;南   &lt;rt&gt;とう&lt;rt&gt;なん   &lt;rtc&gt;たつみ &lt;/ruby&gt;の方角</pre>	
<code>rtc { ruby-position: under; }</code>		<small>とうなん</small> 東南の方角 <small>たつみ</small>

## 1.4. 中付きルビと肩付きルビ

<code>ruby-align: center</code>	<code>ruby-align: start</code>
反 面 の 地 <sub>ち</sub> に	反 面 の 地 <sub>ち</sub> に

### 1.5. 配置位置・文字間のバリエーション

	<code>text-justify: auto;</code>	<code>text-justify: inter-character;</code>
<code>ruby-align: space-around;</code>	永遠のさようならを告げる Dit    adieu	永遠のさようならを告げる Dit adieu
	That's    how    it    happened ここまでが経緯です	That's how it happened ここまでが経緯です
<code>ruby-align: space-between;</code>	永遠のさようならを告げる Dit    adieu	永遠のさようならを告げる Dit adieu
	That's    how    it    happened ここまでが経緯です	That's how it happened ここまでが経緯です
<code>ruby-align: center</code>	永遠のさようならを告げる Dit adieu	
	That's how it happened ここまでが経緯です	

### 1.6. はみ出したルビと前後の文字の関係

	CSS	これは<ruby>羊<rt>ひつじ</ruby>ですか <ruby>狼<rt>おおかみ</ruby>ですか
前後の文字の上にはみ出してよい（詳細は実装による）	<code>ruby-overhang: auto</code>	ひつじ    おおかみ これは羊ですか狼かすか
前後の文字の上にはみ出してだめ	<code>ruby-overhang: none</code>	ひつじ    おおかみ これは羊 ですか 狼 ですか

### 1.7. 改行処理

初期値	<div style="border: 1px solid black; padding: 5px;">           .....この<sup>しゅう</sup>周  <sup>きりつ</sup>期律には.....         </div>
<code>ruby { white-space: nowrap; }</code>	<div style="border: 1px solid black; padding: 5px;">           .....この  <sup>しゅう</sup>周期律には.....         </div>

## 1.8. アクセシビリティ関連のバリエーション

機能	CSS	<ruby>今日<rt>きょう</ruby>
通常	初期値	きょう 今日
スクリーンリーダ等の音声合成の発音に影響がなく、ルビ文字以内の小さい仮名を大きく表示する	rt { text-transform: full-size-kana; }	きょう 今日
ルビ文字を特定した色で表示する	rt { color: red; }	きょう 今日
親文字とルビ文字の間を開く	rt { padding-bottom: 1em}	きょう 今日
ルビ文字を親文字の二分の一より大きく表示する (例：親文字の十分の八)	rt { font-size: 80%; }	きょう 今日

## 2. CSSとHTMLによるルビで対応すべきかどうかの検討中機能

### 2.1. はみ出したルビと前後の文字の関係を細かく指定する

前後の文字の上にはみ出してよい	
前後の句読点のアキの上のみにみ出してよい、他の文字の上はだめ	
前後の文字の上にはみ出してだめ	

### 2.2. 行頭・行末に揃えるかどうか

行頭・行末に揃えない	
行頭・行末に揃えて、前後の文字の上にはみ出さない	
行頭・行末に揃えて、前後の文字の上にはみ出す	