

翌日へ 前日へ

脳log[2013-11-01] Firefoxが句点を行頭に送ってしまうのがあまりに目障りでもう耐えられないので～正規表現(Rubyに劣るECMAScript仕様)～禁則処理(IE完璧。Firefox/Safariに指導)～両端揃え(IE完璧。Firefox満足。Safariを補完)～画像とCSSのDPI

2013年11月01日(金)

「Kindleストアから大手アダルトコミックが大量削除 - 電パブログ」■ちんちんかかもって辞書に載ってる単語だったりする。知らねー。■紙の本でも独善によって一部をなき物にするアマゾンだけど Kindleではそれよりさらに厳しい基準でのぞんでいると。TSUTAYAと一緒に。消えろ(※強勢はなし。付けると願い事みたいになってしまう。平板に、興味のないゴミカスに退場を許可するみたいに発音する)。■ところで、Reader Storeでも Kinoppyでも最初から取り扱いがなかったりするわけですね。終わってるやん。(直販してる)フランス書院が正解。ISBNのついた本を100%カバーすることを目標に掲げた電書書店はないのか？■こんなんのまま紙が廃れたら悲劇だなあ。分を弁えへん私企業に所有させていいもんやないで。こんなところに利益をくれてやるのは同時に人質を差し出すようなもの。あとで後悔することになる。■■■@2013-11-18「あとで後悔」が二重表現の例として挙げられていて、そうかと思って書き直そうとしたがどうもしっくりしない。これは、未来のある時点において後悔することを動詞(の付属部)だけでなく副詞も使って示しているのもあって、つまり、後悔の対象が必ずその時点における過去の事象であることと、後悔するのがいつであるかは別物なので二重表現にはならないのではないだろうか。■「今後悔してる」に対する「あとで後悔する(ことにな

る)」は不自然じゃないと思う。

[tDiary][Firefox][正規表現][javascript] Firefox が句点を行頭に送ってしまうのがあまりに目障りでもう耐えられないので～正規表現(Rubyに劣るECMAScript仕様)～禁則処理(IE完璧。Firefox/Safariに指導)～両端揃え(IE完璧。Firefox満足*。Safariを補完)～画像とCSSのDPI

tDiaryのプラグインとして今回追加したのは auto_nobrの部分だけ。

```

add_title_proc {|date, title|
  auto_nobr = lambda{|src|
    return src.gsub(/^[^{}\\[\]()]*#!'`=:|][、。』!?!?])+[「『(]+
  }
  inline_or_nil = lambda{|src|
    lines = src.split(/\r?\n/)
    return nil if 1 < lines.length
    html = WikiSection.new(auto_nobr.call lines.first).body_to_html
    return nil if html[0,3] != '<p>' or html[-4,4] != '</p>'
    return html
  }
  if title.index('<')
    title.sub(/<span class="title">([<>]+)</span>/){|_0|
      html = inline_or_nil.call(CGI.unescapeHTML $1)
      html ? %/<span class="title">#{html}</span>/ : _0
    }
  else
    inline_or_nil.call(CGI.unescapeHTML title) or title
  end rescue title
}

```

一応。Firefoxの挙動は word-break:break-all が指定された結果である。であるが、だからといって行頭の句点、句点だけの行(ぽつーん)はありえないだろうと思うのです。

やっつけ仕様

1.標準に存在しない<nobr>タグの使用

かといって空白が関係するわけではないから white-space:

nowrap とか使えないし、使えたとして FONTタグを `` に置き換えるようなことに意味を見いだせたのはそれが 2000年頃のことだったからだし、段落全体を一行で表示したいわけでもないのに代替案が見つからない。

2. テキトーなテキスト置換

HikiDocで意味を持ってそうな記号を避けつつ句点 + 1文字を `<nobr>` で困ってる。整形形式でないというエラーが出るパターンがまだ残ってるかもしれない。footnoteプラグインとかわりと文章を渡すから危険だ。

3. 対象がタイトル欄だけ

全体を対象にするなら JavaScriptでやる。でもテキストを対象にしつつタグをインジェクトする方法がわからなかった。

マッチしたテキストノードをドキュメントフラグメント(テキスト+NOBRエレメント+テキスト)と置き換えればよかったのか? 具体的方法が見えたところだが、レイアウトが変わってしまう変更を読み込み完了間際にスクリプトで行うというのはやっぱりよくないかも。

4. 表示を確認したのは Firefox(23.0.1)と IE9だけ

Operaは独自のレンダリングエンジンを放棄したし、Windows版Safariはアップデートが止まったし、Google Chromeはインストール場所がきもいから(今でもそうかは知らない。当然やめるだろうと予想するほどにキモかったということだ)。

「CSS 3におけるテキストの自動改行と禁則処理の定義 - builder by ZDNet Japan」

break-all

任意の位置で自動改行を行うが、日本語のテキストでは「line-break:normal」と指定したときと同じようにゆるい禁則処理を行う。

これを期待して待ってたんだけど。

「アポラボログ: Firefox 15 の禁則処理を修正」

この「word-break」というスタイルシートは文の改行の仕方を指定するもので、もともと Internet Explorer 独自の物だったようなのですが、最近になって Firefox もこれを採用した

らしいのです。そのため、以前の Firefox では無効化されることを想定してこのスタイルシートが指定されていたと考えられます。

サイト制作者の想定としては IE のみにこのスタイルシートを指定したつもりが、意図せず Firefox でも有効化されてしまい、禁則処理のされない読みづらい記事が発生してしまったということだと思います。おそらく、IE の場合はこのスタイルシートが指定されていても禁則処理には影響がなかったのではないのでしょうか。

先行する IEはまとも。後追いの Firefoxはバカ。日本人の貢献が足りないのか？

「Gecko と Webkit の word-break:break-all; ってこれ正しいの? << やおよろグツ!」

良くないと思います。

「word-break: break-all; がW3CでOKになってるし | ぼくんちのバックステージ」

余談:Googleの検索仕様変更について

1ヶ月ほど前から、Google検索の左メニューが無くなっちゃいましたねえ。

スクリプトを無効にしていると今でも出てくるんですよ。置かれたメニューをクリックしなくても大体の選択肢が羅列されるんですよ。フォールバックが機能してるのをほめる前に、スクリプトを使って使いにくくしてることにあきれろ。

@2013-11-02 正規表現パターンについて

auto_nobr処理の中の

```
/[^\{\}\[\]()*"#!'`=:|][、。』!?!?)]+|[「『(]+[^\{\}\[\]()*"#!'`=:|]/
```

というパターンは要するに

HikiDocで特別な意味を持たない1文字+行末禁則文字、または、行頭禁則文字+H

という意味なんだけど、HikiDocで~という部分が^{ちょうぶく}重複してる。パターンの中にパターンを展開する方法を持たない JavaScriptで[?]この重複をどう取り除けるか？

文字列を組み合わせて RegExpのコンストラクタに渡すのは、文字列を Functionコンストラクタに渡すこと(evalと同じ)や、文字列から SQLを組み立てるのに似て好きではない。一面

でわかりにくくなるのを承知で書き直すところなる。

```
/(?=([「『(]+)?([^\{\}\[\]\(\)\*#\!"'\`=:|]))([、。』!?!?])+)?(?:\1\2|\2
```

前半の先読み部分は次のような意味の1～3個のキャプチャを含む。

```
((行末禁則文字)?(HikiDocで～)(行頭禁則文字)?)
```

後半の、対象文字列と実際にマッチする部分はキャプチャを参照するだけの単純な二択。

```
(?:\1\2|\2\3)
```

かっこによるグループ化がないと先読みが最初の選択肢にしかられないことに注意。|の結合が一番弱い。

キャプチャや分岐が増えてるがパターンの繰り返しはないし、トリッキーに見えてもひとたび構造がわかると簡単だと思うがどうだろう。

最終的にこの日記で使用するパターンは ¥3? を付け加えてこうなった。

```
/(?=([「『(]+)?([^\{\}\[\]\(\)\*#\!"'\`=:|]))([、。』!?!?])+)?(?:\1\2\3?
```

これで、かっこで1文字だけを囲った『目』みたいなテキストがひとかたまりとして扱われて折り返されることがなくなる。

ひとつ心配なのは、?によって存在しないことにされたキャプチャを参照することが必ずマッチの失敗を意味するのかどうか。参照が空の文字列に展開されるならば必ず成功すると判断されてもしかたがない。たとえそれが NULLと空文字列の混同だとしてもありそうな話ではある。

「禁則処理がおかしい - Ronten」

ここまで書いてから見つけたのがこれ。

```
word-break: break-all;
から、
word-break: normal;
word-wrap: break-word;
に変更。元々、英数だけの文字がdivをはみ出す現象の防止の為にword-break: break-all;を指定していたが、それだと日本語の句読点が行頭にきてしまうっぽい。
word-break: normal;だけだと、英数だけの文字がはみ出
```

すが、上のように二つ指定すると、日本語も英語も両方うまくいった。

えええええ。くやしいから本文の方だけ CSSで対応する。

```
/* chiffon_leafgreen.css 183行目 */
div.day {
  word-break: break-all;
  word-wrap : break-word;
}

/* に追加して */
div.section {
  word-break: normal;
}
```

思い出した。word-break:break-allと word-break:normal(+word-wrap:break-word)での英字折り返しの違い。normalだとスペースでの折り返しが優先される結果、word-break:break-allしたかった目的

1. 箱の端から端まで文字を満たしたい。
2. 句読点など1文字程度の段差なら許容できる。
3. やりすぎた字間調整(text-justify)は見苦しいからやらない。

が実現できないのだった。これは word-wrap:break-wordでは補えない。

```
a ab abc abcd abcde abcdef abcdefg abcdefgh
abcdefghi abcdefghij abcdefghijk abcdefghijkl
abcdefghijklm abcdefghijklmn abcdefghijklmno
abcdefghijklmnop abcdefghijklmnopq
abcdefghijklmnopqr abcdefghijklmnopqrs
abcdefghijklmnopqrst abcdefghijklmnopqrstu
abcdefghijklmnopqrstuv abcdefghijklmnopqrstuvw
abcdefghijklmnopqrstuvwxy abcdefghijklmnopqrstvwxyz
```

方針は変更せず、タイトルではきっちり右端での折り返しを優先し、本文では禁則処理を優先しよう。そもそもは word-break:break-allで禁則処理が行われるのが本場で、それこそが望みの結果なのに。

@2013-11-04 朝令暮改

「[2.テキストなテキスト置換](#)」が予想以上にプロブレマティ

ックだった。二重ブラケットで囲った URL 中の ? を NOBR タグで囲おうとして URL を破壊し XHTML を破壊していた。既
に書いたように footnote プラグインに引数として渡す文章に
NOBR タグを挿入する問題もある。タイトル欄で footnote プラ
グインは使わない(セクション末尾に脚注を挿入しようとして
も無理だから)が、同じような問題が続出するというのだ。

というわけで、目途をつけておいた JavaScript での実装に切
り替えた。

```
// Firefox(ver.15-23現在まで)が word-break:break-all で禁則処理を
// してくれないので NOBR タグで強制的に特定の折り返しを禁止する。
function auto_nobr(textNode)
{
    var create_nobr = function(text){
        var d = textNode.ownerDocument;
        var nobr = d.createElement("nobr");
        if (text) {
            nobr.appendChild(d.createTextNode(text));
        }
        return nobr;
    };
    var m, re = /[「『(]+.[)]」*|.[、。!!??)』」 ]+;/;
    while (m = re.exec(textNode.nodeValue)) {
        /* assert 0 < match.length in case of infinite loop.
        nobrText = textNode.splitText(m.index);
        textNode = nobrText.splitText(m[0].length);
        nobrText.parentNode.replaceChild(create_nobr(m[0]),
        }
    return textNode; // the last Node of splitted textNodes.
}

function apply_auto_nobr_recursively(node)
{
    var except_tags = {
        "textarea": "タグの包含が許可されていないのか<nobr>で囲っ
        "nobr"      : "二重適用防止"
    };
    for (var child = node.firstChild; child; child = child.nextSibling)
        if ((child.tagName||"").toLowerCase() in except_tags
            // skip blacklist-ed elements.
        ) else if (child.nodeType == Node.TEXT_NODE) {
            child = auto_nobr(child);
        } else if (child.firstChild) {
```

```

        apply_auto_nobr_recursively(child);
    }
}
}

var h2 = document.getElementsByTagName("h2");
for (var i = 0; i < h2.length; ++i) {
    var root = h2[i].parentElement;
    if (! root || root.tagName.toLowerCase() != "div" || -1 == (
        continue;
    })
    // now root is a div.day.
    apply_auto_nobr_recursively(root);
}

```

それから、

ひとつ心配なのは、?によって存在しないことにされたキャプチャを参照することが必ずマッチの失敗を意味するのだろうか。参照が空の文字列に展開されるなら必ず成功すると判断されてもしかたがない。たとえそれが NULLと空文字列の混同だとしてもありそうな話ではある。

と書いておいた懸念は現実のものだった。Firefox(23.0.1)でも IE9でも次のスクリプトは true(マッチした)を表示する。(ちなみに ruby1.8は nil(マッチ無し)を返す)

```
alert(/(?=(A?))\1/.test("B"));
```

これと区別できた方が応用が広がるのに。

```
alert(/(?=(A?))\1/.test("B"));
```

@2013-11-05 Safari(5.1.7)の微妙に異なる振る舞いとやり残し。

Safariでは <nobr> で囲った直前の文字までがひとかたまりになるのか、句点 + 2文字が次行に送られている。読みやすくなるがなんでこうなる？ Safariのユーザー・エージェント・スタイルシートに nobr{white-space:nowrap}がある。これを normalで上書きすると NOBRタグによる禁則が無効になる。空白による分かち書きを行わない日本語文章において white-space指定は空白だけを対象にしたものではなかったか。そういえば全角文字だけの文章と全半角混在の文章で Safariの処理が異なるという報告もあった。ま、<NOBR>でも white-spac

e:nowrapでも、Safariが対象要素の外側にある1文字を余分に巻き込む理由にはならないとおもうけど。パターンから行末禁則文字手前の1文字にマッチするドットを取り除くとSafariの挙動に対応した禁則処理が行えるが、そうするとFirefoxで行末禁則が効かなくなる。おかしなSafariに合わせたりはしない。

既にタグで囲まれているテキストと交差するように <nobr>で囲うことはできない。ということは、「<a>テキスト」とか <a>テキスト。みたいなよくあるマークアップ済みテキストに禁則処理を適用できないということだ(むしろこれに対処したくてSafariは直前の文字(要素)を巻き込んでるのか?という疑いもわいてきたが、直前の文字ならぬ直前の要素にくっついたりはしなかった。残念Safari残念)。これは後付けスクリプトでなんとかできるとは思えない。Firefoxの word-break: break-all完全対応待ち。

@2013-11-06 両端揃え

1.箱の端から端まで文字を満たしたい。

を徹底するためにスタイル指定を足して完成。

```
.day .section, .day .title > * {
  text-align: justify;           /* 両端揃え */
  text-justify: inter-ideograph; /* 日本語両端揃え(IE向けに字間の調整方)
}
```

Firefox(23.0.1)とIE9は期待通り。Safari(5.1.7)は英字手前のスペースだけを使って両端揃えをしようとして英字混じり文が不自然に分断される。これは

3.やりすぎた字間調整(text-justify)は見苦しいからやらない。

に反するけど、Safariだけの問題なので知らない。時間が解決するでしょう。

<追記@2013-11-11>WebKit向けに面白いことをしてる人がいますね。「【目指せePub出版】Webkitでtext-align:justifyに挑戦する | 高橋文樹.com」俺だったら表示されない空白SPAN要素を挿入するんでなくUnicodeの幅0スペースのどれかを挿入するかな。結果が期待通りになるかは確かめる必要があるし、いずれにしろコピペがひどいことになりそうでコンテンツを改変しない方策が望まれるが。</追記>

<追記@2013-11-12>こういう指摘「iOS5のMobile Safari

iでは、日本語でも両端揃えができるようになりました | BALL OG] もあるが、提示された例から判断する限り、iOS4で行われていた「ィ」「ー」の禁則処理が iOS5で行われなくなった結果行末が揃っただけに見える。字間調整が行われた結果ではなく、むしろ退化してないか？それから、全角文字だけで文章を書くならスペースに頼らない字間調整が行われるとの報告もある。</追記>

@2013-11-07「CSS Text Module Level 3 (www.w3.org)」

text-justify

W3C Last Call Working Draft 10 October 2013

The following features are at risk and may be cut from the spec during its CR period if there are no (correct) implementations:

- the 'text-justify' property

Value: auto | none | inter-word | distribute

先月出た文書。text-justify(とその他いくつか)が名指しで消滅の危機。後がない。inter-ideographに代えて distributeを指定しといた方が先々有効かも(属性自体が消えてなければ)。

word-break, line-break

word-breakに line-breakを統合するのはなくなったんだろうか。line-breakの項目がある。うまくないとは思ってたので歓迎。でも句読点の禁則が line-breakの3つのレベル:strict-normal-looseのどこかわからない。

句点は U+3002;CL # IDEOGRAPHIC FULL STOP。かぎカッコ(開)は U+300C;OP # LEFT CORNER BRACKET。UAX #14 (www.unicode.org)では CLも OPも言及されてるけど、CSSの 5.1. Line Breaking Detailsではそれらを含まないクラスに限定して UAX #14を参照してるから CLも OPも位置づけが明確にならない。要は、line-break属性で具体的に挙げられてるのは最低限の要件であって、その他の線引きは User-Agentまかせということだった。書いてありました。

CSS distinguishes between three levels of strictness in the rules for text wrapping. The precise set of rules in effect for each level is up to the UA and should follow language conventions. However, this specification does requi

re that:

将来組み版向けに細かい制御が必要だろうとも言ってるが、当面満足できないということは、line-breakを緩くしておいて必要な部分に<nobr>というアプローチもなくはないのか、な？

KADOKAWAが EPUBの取り扱いに関する文書を公開していたな、とダウンロードしてみた。→「[KADOKAWA-EPUB PORTAL](#)」

■ 3点リーダや2倍ダシがつづく際の禁則の抑制について

3点リーダや2倍ダシ、ナカグロなどは、前後の文字と分離禁止禁則が行われるため、あまり長いと直前の文字から改行されて、意図せぬ表示になることがある。そのようなときは、4文字以上連続する場合を目安に、以下のように「word-break-break-all」を該当箇所にものみ指定すること。未対応のRSもあるが、指定があつて表示が崩れることはないので挿入しておく。

【参考】分離禁止される可能性の高い文字について

CSS3の「line-break」の項には、行頭・行末で前後の文字との分離が禁止になる文字についての記載がある。詳細は「<http://www.w3.org/TR/2012/WD-css3-text-20121113/>」の「line-break」の項を参照。

実際には、まだ禁則処理についてはRS次第であり、line-breakの指定が反映されることも期待できないので、これらに依存するような記述は避ける。

制作仕様ver.1.0.1から引用

基本はword-break:normalで、必要に応じて一部をword-break:break-allで包む^{くるむ}と。どっかで見たような文書(だがちょっと古い)へのリンクもあるぞ。

このPDFでも見たし、数日前に「[自炊ePubのためのあれこれ覚え書き - 道具眼日誌:古田-私的記録](#)」でも見たのだけど、tcyというクラス名を。やっぱりアレをタテチュウヨコと読むんだってことだね。検索したら例によってWikipediaが一番。「読みは「たてちゅうよこ」であり、「たてなかよこ」ではない(JIS X 4051で規定)」。腹腔といい縦中横といい、ローカルルールでもこうと決めたからには正解！みたいなものってどうなん？

@2013-11-09 ECMAScript(3rd&5th ed.)の、Rubyとは異なる、残念正規表現仕様

20131101p01.08の最後の方に書いてたこと。Firefoxでも

IEでも Safariでも同じ挙動——?をキャプチャの中に付けても外に付けても trueを返すこと——を示すので ECMAScriptとして規定されてるんだろうと探してみた。以下該当部引用。太字強調は失われたり付け足したりしてます。

The form (?! Disjunction) specifies a zero-width negative lookahead. In order for it to succeed, the pattern inside Disjunction must fail to match at the current position. The current position is not advanced before matching the sequel. Disjunction can contain capturing parentheses, but backreferences to them only make sense from within Disjunction itself. Backreferences to these capturing parentheses from elsewhere in the pattern always return undefined because the negative lookahead must fail for the pattern to succeed. For example,

```
/(.*?)a(?!(a+)b\2c)\2(.*)/.exec("baaabaac")
```

looks for an a not immediately followed by some positive number n of a's, a b, another n a's (specified by the first ¥2) and a c. The second ¥2 is outside the negative lookahead, so **it matches against undefined and therefore always succeeds**. The whole expression returns the array:

```
["baaabaac", "ba", undefined, "abaac"]
```

Microsoft Word - Ecma-262.doc / Standard ECMA-262 3rd Edition - December 1999 / 15.10.2.8 Atom / 139ページ から引用

こちらでも "always succeeds"と書いてある。

Informative comments: An escape sequence of the form ¥ followed by a nonzero decimal number n matches the result of the nth set of capturing parentheses (see 15.10.2.11). It is an error if the regular expression has fewer than n capturing parentheses. If the regular expression has n or more capturing parentheses but the nth one is **undefined** because it hasn't captured anything, then the backreference always succeeds.

15.10.2.9 AtomEscape / 140ページ から引用

残念だ。thereforeとかあっさり書いちまいやがって。それはまったく自明ではないぞ。

空パターン(※JavaScriptは Perlと違ってダブルスラッシュがコメントになってしまうので作るのに小細工が必要)が必ず

成功するのはわかる。でもそれに対応するのはキャプチャが空文字列を保存していてそれを後から(その文字列そのものにマッチする)パターンとして参照した場合であって、参照すべきキャプチャ・参照すべきパターンが存在しない(=undefinedである。空文字列ではない)ときは必ず失敗して欲しかった。理由はすでに書いたように、この二つを区別できなくなるのが困るからだ。

```
/(?=(A?)\1/.test("B"))
/(?=(A?)\1/.test("B"))
```

もっと実際的な不利益は 20131101p01.06に書いた書き換えが通用しないことだ。これって C++11にも影響する(してる)んでしょ? *SIGH*

@2013-11-12 Safari(Win版5.1.7)で満遍なく字間調整

Safariでは全半角混在文章に対しては自動的に text-justify: inter-wordに相当する字間調整が選択されるらしく、日本語文章にわずかに含まれる空白が過剰に引き延ばされた見苦しい表示になってしまうことをもう書いた>20131101p01.10。

実は word-break:break-allを指定しているとほとんど任意の場所で折り返しができるので(Safariの場合禁則処理も行わないので)わずかなスペースに調整のしわ寄せがいつでも問題にならない(といっても切り落としたような右端のラインは得られないが)。でも禁則処理を施した場合 Safariはなぜか3文字を次行に送ってしまうので(20131101p01.09)、全角1文字分以上の空白が文章を分断してしまうのが問題になる。

どうするか。これに対処して「 」を挿入した人が唯一見つけられる。visibility:hiddenを指定した空白なら挿入してもコピペに影響しないのを Safariと IEと Firefoxで確認した(※)ので自分はこうした。※.textContentには影響するかも。innerTextには影響しないかも。

```
// Windows版Safari(5.1.7)の字間調整は全半角混在文章でスペースに対してしか
// 不可視の空白を挿入することでさらなる字間調整ポイントを Safariに対して教
// 相当うざい結果になる(なにせほぼ任意の2文字の間に SPAN要素が挿入される)
function textfunc2_safari_whitespace_distribution_inter_ideograph(text)
{
    var _xp, new_xp = function() { // XP = expansion point
        if (! _xp) {
            var d = textNode.ownerDocument;
            _xp = d.createElement("span");
        }
    };
}
```

```

        _xp.style.fontSize      =
        _xp.style.letterSpacing = "0px";
        _xp.style.visibility   = "hidden";
        _xp.appendChild(d.createTextNode(" "));
    }
    return _xp.cloneNode(true);
};
var p = textNode.parentNode;
var m, re = /\S(?=\S)/;
while (m = re.exec(textNode.nodeValue)) {
    /* assert 0 < match.length in case of infinite loop.
    textNode = textNode.splitText(m.index + m[0].length)
    p.insertBefore(new_xp(), textNode);
}
// ついでに、Safariが <nobr>の直前直後の2文字を接着して字間調整
try { if (textNode.nextSibling.tagName.toLowerCase() == "nob
    p.insertBefore(new_xp(), textNode.nextSibling);
} } catch(e) {}
return textNode; // the last Node of splitted textNodes.
}

```

スクリプトの全体はページのソースを参照のこと。結構重い処理なのでページの表示後しばらくして文字が移動するのが見えてしまうかも。でも Safariだけの問題なので(略)。

Safariで表示したこのページでテキストを選択してみると、文字と文字の間に白い縦線が入ってるのが見えるんじゃないだろうか。それが visibility:hiddenなスペースだと思われる。行末から行頭に向かって1文字1文字の間に1ピクセルの字間を配ってるかんじ。行頭に着いても分配する字間が余ってる場合は2ピクセル目をまた行末から配る、と。たまに2文字くっついたままになってるのは禁則処理のために挿入した NOBR(開き)タグの直前と直後の文字。Safariはなんでこの2文字を接着してしまっただけで字間調整も折り返しも行わないんだろう。<NOBR>の直前に字間調整用の SPANを挿入するとめでたく接着が解けたが、SPAN挿入の二重適用を避けつつ NOBRに限ったアドホックな処理を追加せずに済む方法は……。<追記@2013-11-14>上のスクリプトは Safariを特定したものなのだし、NOBR要素狙い打ちで字間調整用の SPANを注入することにした。</追記>

「W3C日本語組版ノートとCSS3 - JAGAT」

WebKit以外は禁則に対応している。CSSには禁則ルールを

「通常」から「厳しくする」と3段階の設定がある。両端揃えについては実装依存、「どういう実装をしてもいい。ただ日本語ではJLREQを参照するといい」としている。EPUB仕様も同様である。

Firefoxで `word-break:break-all`を指定すると禁則処理が行われなくなるのが問題。

Safariが全半角混在文章に対して日本語向けの両端揃えを選ばないのが問題。

「public-html-ig-jp@w3.org Mail Archives」

@2013-11-14 PS3・泥ケー・Opera

PS3のインターネットブラウザも今では WebKitベースだとか(実際そう名乗っていた。診断くん)。なかなかきれいなフォントレンダリングで(でもズームするとややとする)、禁則処理、字間調整の方法まで含めて Safariによく似た結果。スクリプトを有効にすることでこのページも期待通りの表示になった。

もうひとつブラウザ。ドロケー(URBANO L01)で表示すると中心線近くでせせこましく折り返したり折り返さなかったりする現象(※)が見られたんだけど、google/京セラのせいにして放置していいのか、DPIの高さにこのページが対応できていなかったりするのか。

※スクリプト実行中は画面に見えてる3日分のタイトルがせせこましく折り返してた。実行後は先頭の1日だけがせせこましく折り返したままだった。

また、ナビゲーションリンクをクリックすると再現性なく HTTPステータスコード 505 HTTP Version not supportedが返ってくることがわずかな時間に何度もあった。やっぱりブラウザ(GCかどうか知らないが)が信用ならんのかな。LTE経由やったから auがいらん茶々入れてたんかな。

Opera17も試してみたんだけど、フォントの設定がないのんな。開発者プレビューにはあったみたいだけどその場所にもなかった。閲覧者が読みやすいと感じるフォントは閲覧者が知っている(そして設定している)だろうと思って HTML/CSSではフォントファミリを指定してないんだけど、その結果がMS Pゴシックのビットマップフォント。3段階ほど文字を大きくするとベクタデータに切り替わるのか見られる表示になるけど、MS Pゴシックをあえて指定してすら文字が美しい Safariとは雲泥の差。見るに堪えない。

@2013-11-15 テーマ画像。DPI

URBANOのブラウザは Google Chromeではなかった。バージョンが 4.XXXだということがかろうじてわかったがまったく素性が知れない。これは不自由ゆえか不慣れゆえか。

スマホの Google Chromeで表示してみると PCで見るより一行の文字数が少ない。固定サイズの背景画像を基準にして横幅をピクセルで指定しているために、ひと文字ひと文字をより多くのピクセルを使って精細に描画する(高DPIってこと)スマホでは文字数が減るんだろう。こういうのってベクタ画像をスケールしたり、左上角+辺の繰り返し+右上角の画像3枚構成で横幅を可変にしたりするのだろうか。そういうことができるのかどうか知らないけど。それかこういうときにこそ画像の DPI値(JPEGにそういう値がなかった?)が役目を果たしてブラウザが勝手にリサイズしてくれたりするんじゃないだろうか。DPIとピクセル数で現実世界の長さがわかるわけで、それを共通の尺度にしてブラウザが再度スマホにおいて画像を表示するのに使うべきピクセル数を求められるはずでは?でも現実世界の長さを共通の尺度にするとスマホの画面サイズはPCのモニタと比べて小さすぎるのか。でもそれはスマホが DPI設定をそれなりの高さに抑えておいてズームを駆使すればいいだけのことじゃないか。Windowsの DPI設定がモニタのピクセル密度を全然反映していないのだから何を考えても机上の空論か。そもそも、日記のテーマ(chiffon_leafgreen)で使われている画像が DPI値をセットできない GIFだった。画像にだけ DPIが指定してあっても、というのもある。画像のピクセル数を基準にして div.day要素の横幅が 510pxと指定してあるから、こちら、画像が何ピクセルを使って表示されるのかに合わせて変換されないとつじつまが合わない。そのへんはまた CSSの px単位が論理的/相対的な単位であるとかが関係してきて……。これかな? 「'px' pixels; 1px is equal to 1/96th of 1in」 DPIを固定して定義されてるから pxと現実世界の長さがいつでも 1:1 対応する、その結果 CSSの 1pxとモニタの画素とは 1:1 対応しない、と。テーマ(※ライセンスはGPL)で使われている画像を 96×96DPI指定した $P \quad N \quad G$ pingと同じピンではないの? に変換するだけで解決するといいなあ(儚い希望)。

@2013-11-19「HTMLで文字詰めするタイポグラフィー用JS | fladdict」

これは字詰め。禁則とも字間調整(広げる方)ともかち合うので、これを適用しようとするするとタグの交差に対処する方法を真剣に考えないといけない。でもやりたいなあ。読点とかぎ括

弧、中黒は全角だと間延びしすぎだし、だからって半角文字を使うのも違う気がするし。ああでも、フォントを知らないとめくらうち盲撃ちになってしまうのか……。

@2013-11-20 字詰め。

やった。ほとんど昨日貼ったリンク先のスクリプト(FLAuto Kerning.js)の移植。行頭の約物の処理だけ省いた。折り返しでの天付きをせずにそれだけするのもどうかと思って。いい加減長いので詳細はソースの textfunc4_jizumeへ。

問題点

- さらに重くなった。重たい字間処理と重なる WebKit系が悲惨。
- MS Pゴシックだと詰まりすぎる。つまり Opera18で詰まりすぎる。
- Safariでは一部の記号の一部が元々詰まっていて、そこに字詰め処理が入ると詰まりすぎる。
- 適用先の要素に 0以外の letter-spacingが指定されていた場合は、それに加算した letter-spacingを設定すべきではないか？

まとめると、フォントにメイリオを指定した Firefox(25.0)、Internet Explorer(9)以外お断り仕様になってしまいました。

Firefox25での表示(ベンチマーク)> [20131101p01.pdf](#)(610KiB)

* 全角文字の字間は拡大するが英字はそのままなので、一部は間延びして一部は詰まって見える。全体に散らした方が目立たないのに。@2013-11-19 Safariと同じ方法をより限定された対象に適用するだけで矯正できるのでやってしまった。対象を限定しなくても副作用はないけど処理負荷を考えて限定した。コピペコード率高し。

? @2013-11-09 勘違いしてたけどこの部分は Rubyなのでパターンの中にパターンを埋め込む方法で繰り返しをなくせる。それに後に明らかになったように、この後の書き換えは各種ブラウザの JavaScriptで無効な、Rubyでしか使えない方法だった。ECMAScriptとしてあまりありがたくない動作が規定されてるのかも。

最終更新: 2013-11-21T07:54+0900

See also...

- カテゴリ[javascript]の他の日記
 1. 2013年11月 1日p01 [tDiary][Firefox][正規表現] Firefo

- xが句点を行頭に送ってしまうのがあまりに目障りでもう耐えられないので～正規表現(Rubyに劣るECMAScript仕様)～禁則処理(IE完璧。Firefox/Safariに指導)～両端揃え(IE完璧。Firefox満足*。Safariを補完)～画像とCSSのDPI
2. 2011年11月 9日p01 [本] アマゾンの URLに含まれる ISBNっぽい数字を紀伊國屋書店BookWebで電子書籍検索するブックマークレット。
 3. 2011年 9月16日p01 trailing commas ,
 4. 2010年11月25日p01 アマゾンの URLに含まれる ISBNっぽい数字をジュンク堂で検索するブックマークレット。
 5. 2010年 1月31日p01 JSLint
- カテゴリ[Firefox]の他の日記
 1. 2013年11月 1日p01 [tDiary][正規表現][javascript] Firefoxが句点を行頭に送ってしまうのがあまりに目障りでもう耐えられないので～正規表現(Rubyに劣るECMAScript仕様)～禁則処理(IE完璧。Firefox/Safariに指導)～両端揃え(IE完璧。Firefox満足*2。Safariを補完)～画像とCSSのDPI
 2. 2013年 2月 9日p01 クラッシュに備えてアドオンリスト。
 3. 2010年 8月28日p02 plugin-container.exe
 4. 2009年 7月 3日p02 [正規表現] XRegExp-1.0.1と Firefoxの/(?!)/
 5. 2009年 6月28日p01 [正規表現] .NETの正規表現の再帰について(20080111p01の補足)。
 - カテゴリ[tDiary]の他の日記
 1. 2013年11月 1日p01 [Firefox][正規表現][javascript] Firefoxが句点を行頭に送ってしまうのがあまりに目障りでもう耐えられないので～正規表現(Rubyに劣るECMAScript仕様)～禁則処理(IE完璧。Firefox/Safariに指導)～両端揃え(IE完璧。Firefox満足*3。Safariを補完)～画像とCSSのDPI
 2. 2012年11月28日p01 Wikiスタイル(hikidoc.rb)にルビを。(impl_ruby_markup.rev1.1.patch)
 3. 2012年 1月15日p01 SourceForge.net: tDiary: TDiary::Plugin の多重初期化を減らしたい
 4. 2011年10月 4日p01 リファラ置換リスト(対google)晒し。
 5. 2011年 8月 3日p01 PageDownキーによるスクロール量をごによごによるプラグイン。
 - カテゴリ[正規表現]の他の日記
 1. 2013年11月 1日p01 [tDiary][Firefox][javascript] Firefoxが句点を行頭に送ってしまうのがあまりに目障りでもう耐えられないので～正規表現(Rubyに劣るECMAScript仕様)～禁則処理(IE完璧。Firefox/Safariに指導)～両端揃

え(IE完璧。Firefox満足*4。Safariを補完)～画像とCSSのDPI

2. 2010年 9月 7日p01 [SakuraEditor] 検索。置換。
3. 2010年 7月 9日p01 [SakuraEditor] 「鬼車と bregonigに hitEnd(20100531p01)機能が搭載されることを願う他力本願日記」
4. 2010年 5月31日p01 requireEnd(), hitEnd()
5. 2010年 5月19日p01 鬼車。条件付き選択。

ツッコミ・コメントがあればどうぞ!

コメント:

お名前:

投稿

[翌日へ](#) [前日へ](#)

Generated by [tDiary](#) version 2.3.3.20091124

Powered by [Ruby](#) version 1.8.7-p371