

ADP: IE6のwidth解釈バグ対処法

2004年07月20日 15:11 | コメント(5) | トラックバック(8) |  186 users

CSSをやってはまる典型例は、「CSSせっかく使ってみたのに、横幅がIE6とMozilla Firefoxであわなくてレイアウトがうまくできないんだけど」ということではないか。言い古されているが、ADP流にまとめてみる。

原因は、Internet Explorerのwidth/height（ボックス）解釈に関するバグである。このことを知らない人は、ふつうIE6で見た目を調整してから他のブラウザ（FirefoxやOpera）を見るので、「なんだよずれてるじゃないか」と怒り出す。よくある話。

ADPでの結論を先に書く。**widthとpadding/borderを一緒に指定しない、特に、ピクセル単位で合わせるところではborderを使わない。**これに尽きる。

IE6では、IE5.5までに存在したいくつかの致命的な解釈バグを直した「標準準拠モード」と、間違った解釈のままつっぱしる「後方互換モード」を、ページ制作者が選択することができる。実際には、HTMLのDOCTYPE宣言の有無や内容によってブラウザがどちらのモードで解釈するかを決定する。この「DOCTYPEスイッチ」は、OperaやFirefoxにも存在する。どういうDOCTYPEのときにどっちになるか、を覚えるのは大変である。

一番の問題は、IE6はHTMLの**いちばんあた**まにDOCTYPE宣言を書かないと必ず後方互換モードとして解釈してしまうことである。よくあるケースは、XHTMLで書く場合。通常、XHTMLではDOCTYPE宣言の前にXML宣言を書くため、IE6では有無を言わせず後方互換モードになってしまう。ページ制作者が標準にあわせて書いても、IE6が後方互換で解釈してしまうと思ったとおりに表示されないし、IE6の後方互換の見た目に合わせて書けば、標準準拠で解釈するFirefoxやOperaで思ったとおりに表示されない。はまる。

このひとつの回避方法は、XML宣言を書かないことである。こうすれば、1行目がDOCTYPE宣言になり、IE6も標準準拠モードで解釈してくれる。Movable Typeのデフォルトテンプレートをはじめ、この回避方法が一般的だろう。

ただし、ADPでお奨めするやり方はちと違う。制作者が標準に従ってXHTMLを書いて、XML宣言もちゃんと書く。これだと、IE6は後方互換モードで解釈して、FirefoxやOperaは標準準拠モードで解釈するわけだが、ADPのやり方ならばまったく同じ表示になる。つまり、IE6で標準準拠モードで解釈させるのをすっぱり諦めて、後方互換でも標準準拠でも同じように表示させるように工夫するわけだ。

このメリットは、なにより、標準準拠で書けること、後方互換モードしかないIE5.5でも（たぶん）無事に表示されること、ブラウザがどのモードで解釈するかをいっさい考えずにすむことである。

ここでは、widthのみについて書く。その他、標準準拠と後方互換のおおまかな話は、[文書型宣言とブラウザでの表示 >> http://www.mars.dti.ne.jp/~fuming/advanced/dtd.htm](http://www.mars.dti.ne.jp/~fuming/advanced/dtd.htm)（訪問者に優しいWebサイト作り）がわかりやすい。

以下解説。

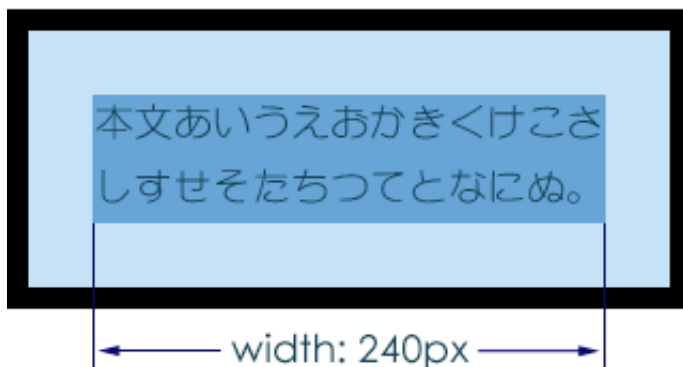
まず、width解釈に関する説明を

```
.box {  
    width: 240px;  
    padding: 30px;  
    border: 10px solid black;  
}
```

の例でやってみる。

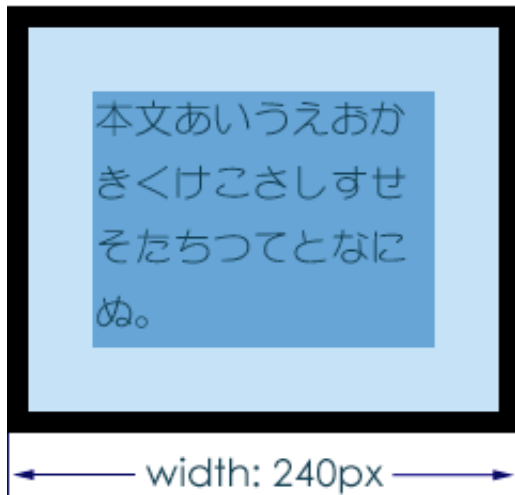
正しいCSS解釈におけるwidthは、図1のようにcontent（中身）の幅であり、paddingやborderはその外側に置かれる。上記の例の場合、borderまでを含めたボックス全体の幅は320ピクセルになる。

図1：正しい解釈



間違ったCSS解釈におけるwidthは、図2のようにborderまでを含んだボックスの横幅である。後方互換モードでは、こちらに従ってwidthを解釈する。

図 2: 間違った解釈



これを知らずにIE6（の後方互換モード）で見た目を調整すると、他のブラウザではボックスが大きくなってしまい、はみでたり、floatが崩壊したりする。

では、図1を、標準準拠な書き方で書いて、後方互換モードでもそのまま表示されるようにするにはどうするか。

解法。まず、大きなブロック要素を用意し、図3のようにwidth: 320pxを指定し、paddingは指定しない（=0にする）。

```
.boxA {  
    width: 320px;  
}
```

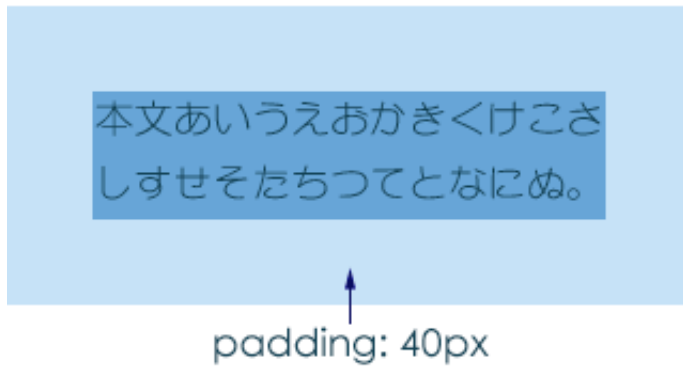
図 3: 外のボックスで width を指定



その中のブロック要素で、図4のようにpadding: 40pxを指定し、widthは指定しない（=autoにする）。fontなどの書式は、ここに指定する。なお、paddingの代わりにmarginにする場合は、[ADP: マージンの相殺と背景 >> http://adp.daa.jp/archives/000574.html](http://adp.daa.jp/archives/000574.html)の話も気をつけなければならないので、paddingにしておいたほうが安全だ。

```
.boxB {  
    padding: 40px;  
}
```

図 4: 中のボックスで padding を指定



こうすれば、contentの幅はどちらのモードでも結果的に240ピクセルになる。大事なことはwidthとpadding/borderを一緒に指定しないことである。

えー、これじゃあ枠線がなくなっちゃうじゃん、と思うかもしれないが、必要ならばどちらかのボックスで背景画像を使ってborderを表現する。たとえば、左右は幅固定がいいけれど、上下は可変、だったら、上下にはborderを使ってもいいが、左右のborderは背景画像で代用する。

中のボックスにならborderをつけられるんじゃないか、と思っても、中のボックスにborderと背景画像をつけようとすると、[ADP: ピクセル単位で合わせる場所ではborderを使わない \(IE7でも!\)](http://adp.daa.jp/archives/000571.html) >> <http://adp.daa.jp/archives/000571.html>で説明したように、背景画像の開始位置がIE6/7と他のブラウザで変わってしまう、という問題が出てくるので注意が必要だ。というより、ピクセル単位で合わせるべき場所ではborderを使わない、と覚えてしまったほうが楽だ。

2つブロック要素を用意するとなると、div要素の乱発になるのではないかと心配する人もいるだろう。しかし、たとえば、ul/li要素でwidthを指定しli要素でpaddingを指定する、dl要素でwidthを指定しdt/dd要素でpaddingを指定する、body要素はwidth: 100%のままでp要素でmarginを指定する、などとすれば、div要素をいっさい使わなくてもデザインが可能である。実際、現在のADPはそれでやっている。

お試しあれ。

トラックバックURL: <http://adp.daa.jp/cgi/mt2/mt-tb.cgi/264>

Trackbacks:

いい加減なIE6.0の文書型宣言 >> <http://digibot.jp/blog/archives/2004/08/ie60.html>(WEBFACTORY :: BLOG): 2004年08月03日 04:57

これでもかというくらいIEの文書型宣言について探し回った結果、IE6.0のDOC...

Doctypeに関する注意事項 >> [http://www.stylesheet-stylebook.com/wiki/index.php?](http://www.stylesheet-stylebook.com/wiki/index.php?Doctype%A4%CB%B4%D8%A4%B9%A4%EB%C3%ED%B0%D5%BB%F6%B9%E0)

[Doctype%A4%CB%B4%D8%A4%B9%A4%EB%C3%ED%B0%D5%BB%F6%B9%E0](http://www.stylesheet-stylebook.com/wiki/index.php?Doctype%A4%CB%B4%D8%A4%B9%A4%EB%C3%ED%B0%D5%BB%F6%B9%E0)

(PukiWiki/TrackBack 0.1): 2004年08月03日 13:13

特にIEに多い問題ですが、Doctypeをきちんと書かないとCSSを上手く解釈してくれないなど、様々な問題が発生します。そのあたりを詳しく解説しているサイトをリストアップ。

IE6のwidth解釈バグ対処法 Activating the Right Layout Mode Using the Doctype Declaration Doc...

スタイルシート修正 >> http://www.heartfield-web.com/item_208.html(Heartfield): 2004年09月05日 01:13

MTの時はMacのIE以外はなんとか同じ雰囲気が表示されていたのですが、Nucleus移行中にWinのIEで左と右のメニューがズドンと下に落っこちてしまいました。ログを見るとやはり圧倒的な数のWinIE。多くの人々.....

HTMLの文書型宣言 >> <http://blog.livedoor.jp/kyorecoba/archives/7262014.html>(パンパでガウチョ): 2004年09月27日 00:23

えーと、それこそ素人なので、大それたことを書こうとしているわけではありません(笑) ちゃんとしたことを知りたければ、検索でもしてお調べください。「文書型宣言」、DTD (Document Type Definition)なんて略称されます。具体的に大体どういうものかという、この...

width解釈の違いへの対応策 >> http://moso.upper.jp/blog/archives/2004/11/width_1.php(今日の覚え書き): 2004年11月24日 23:06

最近、「ブラウザによって見え方が違うんだよ(つд)`)」と年中ごねておりましたが、どうか問題が解決したので、このことで二...

バグ再び...IE6のwidth解釈 >> <http://carnival.fool.jp/blog/archives/2004/12/ie6width.html>(Truth Is No Words): 2004年12月27日 11:14

先日から、BROWN.Hさんがコメントで教えて下さったHTML文書の文法チェック... スタイルシートの落とし穴 >> http://blog.nobu-nobu.com/archives/2004/12/post_18.html(禁煙貯金ブログ): 2004年12月30日 19:34

。(`д´)。当ブログでテンプレートを公開していますが、これがIE6以...

Win ie5x対策 幅を指定したら余白は0に >> <http://blog.y-iweb.com/archives/000137.html>(できる！CSSを使いこなす weBlog): 2005年04月02日 01:57

たまたま訪れたサイトで、自分のサイトが紹介されているというのも、なんか不思議な感じだ。IE5.xでのボックス表示を修正する 上記サイト「小粋空間」は、Movable Typeをカスタマイズする時に参考にして、ブックマークしていたものの、その後カスタマイズはほとんど行わ...

Comments:

.....
アサノ: 2004年07月21日 12:38

んがー。私はこの方法を無意識のうちにやり出して現在に至るんですが、こういう事だったんですねー。(お`) きちんとした説明にして下さって嬉しいです。やっぱり説明するという事は大切ですねー・・・。

Momomo: 2004年07月21日 19:15

アサノさんありがとうございます。書くぞ書くぞ、と思ってから2ヶ月ぐらいか

かっちゃんでしたが、がんばって書きました。喜んでいただけてうれしいです。

Momomo: 2004年08月06日 02:10

自分用メモ：中のブロックにborderを指定しても別に問題なさそう（Windowsの3ブラウザで確認）。なんか、borderつけると背景画像の始まる位置がブラウザによって微妙に変わる、みたいな記憶が自分の中であっただけだけど、勘違いだったかも。

Momomo: 2004年08月07日 02:11

勘違い勘違い。ピクセル単位で合わせるべきところではborderを使わない、というのは理由が違うんだ。というわけで、結論としてはborderは使わないほうがいい。理由はのちほど。



Momomo >> <http://adp.daa.jp/profile.html>: 2007年09月21日 20:24

以前の私のコメントのとおり、中のブロックにborderを指定できるかどうかの話は自分自身かなり混乱していましたが、後のエントリで問題点をちゃんと説明できるようになったので、リンクを追加しました。、

正解は「中のブロックでborderを使えるケースもあるが、うっかりしていると落とし穴に落ちるので、ピクセル単位で合わせるべきところではborderを使わない（使えない、ではなく）、と覚えてしまったほうが安全」です。

Copyright 2004-2009 Alternative Design Project by Momomo.