

小国文男

PDF版

QuarkXPress

3.3 ~ 4.1 対応

奮戦記

for Macintosh

私は書籍など縦組みの組版が比較的多く、文字組みはほとんどがベタ組みだ。特別なエクステンションを使わずに、できるだけキッチリしたベタ組みができないだろうか。そしてこれまた行き着いたのは、必要な文字数がピッタリ収まるテキストボックスを作ることだった。二十字ならその二十字がピッタリ収まるボックスを作れば、考え方としてはキッチリとベタ組みになる。考えてみれば超簡単なこと。行揃えにしてもそうだが、このあたりで私は、クオークエクスプレスはアウトにボックスを作ってはダメなんだと考えるようになった。

フォント埋め込み
第3版

えでいっとはうす

第1回●1 万分の 1 ミリの明暗	3
ピッチリサイズテキストボックスの行揃えと文字揃え効果	
第2回●マスターページとプラトニック	9
マスターページアイテムには指を触れるな	
第3回●自由と不自由は紙一重?	15
マニュアルな表組み作業を考える	
第4回●キーワードは「直しやすいデータ」	21
スタイルシートをうまく使おう	
第5回●知っている「ちょっと得する」Tips	27
各種設定に隠れた楽チン操作	
第6回●備えあればトラブルもまた楽し?	33
トラブルシューティング	
索引	39

本書は「Gordian Knot」誌（2000年6月第24号を最後に廃刊）に6回連載した原稿を、同誌の了解を得て再掲したものです。Webでの配信を目的に作成しました。

もともとは私のホームページに1996年から書いている連載で、仕事で毎日使用しているQuarkXPress 3.3Jについて、その体験から、紙面編集に役立つような情報を紹介してきました。情報としてはすでに古くなったものもありますが、この連載では比較的ベーシックな内容を集めています。ささやかな体験ですが、多少ともお役に立てれば幸いです。

扱っているのはバージョン3.3Jですが、最新の4.1Jとも共通する内容に留意しています。その意味で「3.3～4.1対応」としていますが、4.1独自機能については扱っていませんので、あしからずご承知おきください。

再掲にあたり、もともとモノクロだった紙面を一部カラー化しました。第2版ではバージョン4.1を使用し、その新機能を使って索引を作成、巻末に追加しました。PDFはAcrobat 4.0形式で全フォントを埋め込みました。ただし、手持ちの埋め込み可能フォントが少ないため、一部の項目末にある「制作環境」記載のフォントとは異なっています。 2000年12月 小国文男

本書の内容は、アプリケーションソフトのエンドユーザーとしての筆者が自らの体験をもとに記述したもので、Quark社とは直接関係ありません。QuarkXPressはQuark社の、MacintoshはApple社の、Illustrator、Photoshop、Acrobatはアドビシステムズ社の、それぞれ登録商標です。そのほかすべてのブランド名、商標、登録商標は、それぞれの帰属者の所有物です。

第1回

1万分の1ミリの明暗

ピッタリサイズテキストボックスの行揃えと文字揃え効果

Web ページ (<http://ha3.seikyuu.ne.jp/home/ogn/>) 上で私が連載している「QuarkXPress 奮戦記」が上高地氏の目に止まったようで、今回から連載することになった。

私は現在、フリーランスでライター&エディター、そしてDTPオペレーターをしているが、いわゆる手書き新聞からDTPの世界に入っているため、印刷現場の経験はない。QuarkXPressを使い始めて5年目。その間現場の方々にさまざまなことを教えられながら現在に至っている。だから本誌読者の関心に耐えるかどうかはなはだ不安だ。ただWebページ上のそれが、よくぶつかる問題でマニュアルなどにもあまり載ってないことを書いているため、多少の反響があった。そんなわけで、厚くましくも連載をお引き受けした次第。

オリジナルは1996年9月から書いており、すでに2年以上続いている。その間に私自身の認識が変わったり新しい発見があったりしたので、この連載はあらためて全体を見直して再構成しながら書いていくことにしたい。

なお、ここは私自身が組めという上高地氏のリクエストである。私は普段ベタ組みばかりだが、それでもよいということなので、ここだけ他に比べてページのスタイルに違いがある。特に段間スペースが広いのが少し気になるが、まずはご了承をいただきたい。

さて第1回はテキストボックスの話。バージョン3.3でも4.0でも同じだが、サンプルは3.3を使用している。

どうやったら縦の行が揃うねん???

私がQuarkXPressを使い始めて最初にとまどったの

は、縦組み多段組みの行揃えだった。タブロイド判縦型の9段組み新聞紙面が、最初の仕事だった。

当たり前のことだが、縦組み多段組みの紙面は、意図的にずらしたり囲み記事など一部に例外もあるが、一般的には上から下まで行位置が揃っているというのが不文律のようだ。しかし実際にQuarkXPressで作業を始めると、うまくいかない場面に遭遇する。問題は見出しや写真、資料などのために何段かを切り壊す場合、つまり各段でテキスト第1行目の位置が違うケースの行揃えだ。

横組みにはベースライングリッドという便利なものがあるので、段落書式で「ベースライングリッド固定」を設定しておけば、それほど苦勞をすることはない。ところが縦組みではベースライングリッドは使えない。

私はまず、ガイドラインを使って目視で揃えようとした。しかし画面上で揃っているように見えても、プリントすればわずかだがズレが出る。確かにほんのわずかのズレなのだが、紙面ではけっこう目立つ。これでは話にならない。しかし、縦組みの行揃えで苦勞するのは私だけではないらしく、今でも行が揃っていない印刷物を目にする可能性がある。

何かよい方法はないかと思って雑誌などを読んでみると、行位置を揃えることは当然のように指摘されていた。ところが、どうやって揃えればよいかということになると、その方法にはなかなか行き当たらない。かといって、フリーの身では高いXTensionを買う金もなかった。

ちなみに、「Base&BOX」というXTensionを使うと、行ごとにガイドラインを引いてくれる機能がある。試してみると確かに便利だ。しかし画面がガイドラインだら

見出しのために8行空けたが、アバウトに空けたので三段目と四段目で行がずれているのがおわかりだろうか。サンプルは分かりやすいようにややオーバーにすらした。



けになってかなり目障りだし、うっかり触って動かしてしまうと面倒。ガイドを消して作業すればいいのだが、今度はやたらと吸着するのも気になる。どうも今ひとつ、というのが印象だった。

縦組み行揃えのための4点セット

そこで私がいろいろと試行錯誤してたどりついた縦組み行揃えの方法は、必要な行がピッタリ収まるテキストボックスを作ってそれをベースにするというものだ。これには、四つのポイントがある。

■「絶対行送り」を使う

まずは、テキストの段落書式で「絶対行送り」を使うこと。そして、一度行送り値を決めたら、本文組みではそれを変えないことがミソだろう。

QuarkXPressの行送りには、環境設定の「行送りの自動設定」で指定したパーセント値を使う「自動行送り」、固定値を指定する「絶対行送り」、「+」をつけた数値で指定する「増減幅指定行送り」の3種類がある。このうち「絶対行送り」以外は、文字サイズによって行送り幅が可変するからうまく行揃えできない。ついでに「ボックス内の行揃え」を「ジャスティファイ」にするのも、同様にふさわしくない。

■テキストボックスの幅は計算して作る

「絶対行送り」にしていれば、必要な行数が収まるテキストボックスの幅は、次の算式で得ることができる。

$$\text{ボックスの幅} = \text{行送り値} \times (\text{行数} - 1) + \text{文字サイズ}$$

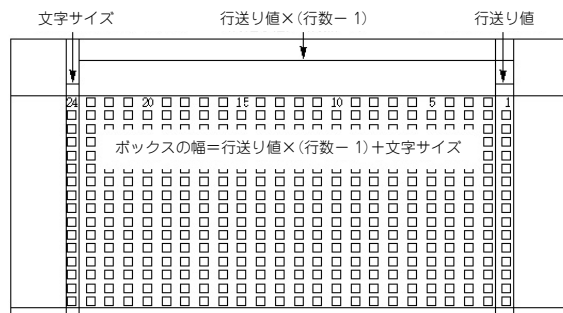
このテキストボックスが左右同じ位置で上下に並んでいるか、あるいは複数コラムのテキストボックスになっていれば、切り壊すためには、そのために必要な行数分だけテキストボックスを小さくするか、同じく必要な行数分の別のボックスを重ねて枠空けをすればよい。これは次の算式で求められる。

$$\text{枠空けサイズ} = \text{行送り値} \times \text{行数}$$

考え方としてはこれで必ず行が揃う。私は文字の単位



絶対行送りは、段落書式の「行送り」フィールドに数値を指定。



テキストボックスの幅の計算。図でわかるように、最終行は文字サイズだけであることに注意。「行送り値×行数」とすると間違いだ。

テキストボックスの設定

左肩の横座標(X): 14.75 mm
 左肩の縦座標(Y): 15 mm
 幅(W): 118.5 mm
 高さ(H): 180 mm
 ボックスの角度: 0°
 ボックスの傾斜: 0°
 角の半径: 0 mm
 コラム数: 4
 コラム間隔: 4 mm
 テキストとの間隔: 0 pt

1行目のベースライン
 オフセット: 0 mm
 最小値: アセント

ボックス内の行位置
 配置: 右揃え
 段落間スペース(≦): 0 mm

バックグラウンド
 カラー: 白
 シェード: 100%

印刷しない

OK キャンセル

テキストボックスとの間隔の設定は「アイテム」メニュー→「設定」で行う。ダイアログの左下にある。

回り込みの設定

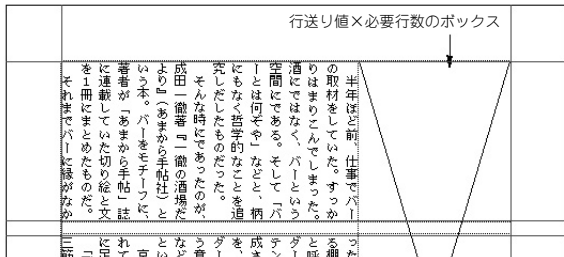
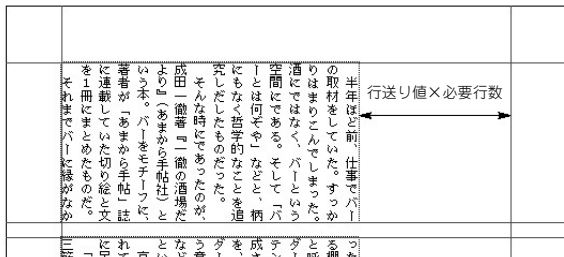
モード: アイテム

上辺: 0 pt
 左辺: 0 pt
 下辺: 0 pt
 右辺: 0 pt

内外反転

OK
 キャンセル

回り込みの設定は「アイテム」メニュー→「回り込み」で行う。



枠空けするには、テキストボックスを必要行数分小さくする（サンプル上）か、必要行数分のボックスを重ねる（サンプル下）。これでパッチリ行が揃う。

にもっぱら級数を使っているが、もちろんポイントでもまったく同じだ。

ただし、QuarkXPress上でこれを実現するには、二つの前提が必要だった。そして、うまくいかなかった時の原因の一つもここにあったのだ。

■ボックスの「テキストとの間隔」を「0」にする

一つは、テキストボックスの「テキストとの間隔」を必ず「0」にすることだ。

「テキストとの間隔」とは、テキストボックスの周囲内側に設定されるスペースのことで、アイテムメニューの「設定」ダイアログに含まれている。デフォルトではここに1ptが設定されていて、このままではいくら先の計算でテキストボックスを作っても取まらない。

またこのスペースがあると、行揃えには常にその数値を勘案しなければならず、作業が複雑になる。「0」にしておくことが、行揃えをずいぶん楽にするわけだ。

■重ねるボックスの「回り込み」も「0」にする

もう一つは、重ねるボックスの「回り込み」値も「0」にすることだ。

「回り込み」は、ボックスの周囲外側にテキストを回り込ませる時のスペースを設定するためのもの。デフォルトはこれも1ptだから、同様に「0」にしておくのが楽だ。これで、ボックスのサイズがそのまま枠空けのサイズとして使える。画像などはその上に貼ればよいのだ。

この「テキストとの間隔」「回り込み」の設定は、ドキュメントを開かない状態で行っておけば、以後に新規作成する全てのドキュメントに適用されるし、データを移動しても全く問題ない。

こうして、①絶対行送り、②テキストボックスは計算して作る、③「テキストとの間隔」が「0」、④「回り込み」が「0」という4点セットでピッタリサイズのテキストボックスを作るようになってから、私はきわめて楽チンに行揃えができるようになった。

■注意がいるのはボックスの右側

もちろん、何でもかんでもテキストボックスはピッタリサイズが必要、というわけではない。大切なのはピツ

タリサイズの版面を作ることで、これはマスターページでの作業。マスターガイドが必要な行の収まるピッタリサイズであれば、中に入るテキストボックスは、「ボックス内の行位置」を「右揃え」にしている限り、右端の位置に注意すればいいというケースが多い。

詳しくはマスターページについて触れる時に紹介しようと思うが、ピッタリサイズのテキストボックスは、そのための基礎と考えていただければいいだろう。

なお私の Web ページ上で、ピッタリサイズのテキストボックスでは、最終行に欧文フォントがあると収まらないという指摘をいただいた。日本語フォントの欧文を使っている分には問題ないが、確かに欧文フォントを使用するとベースラインの違いにより指摘の通りになる。だから注意する必要はあるが、私が問題にしているのは縦組みの行揃えだから、あまり欧文フォントは使わない。

アバウトで作ったらダメなんや

さて、縦組み多段組みの行揃えの次にぶつかったのは文字組みだった。

QuarkXPress の文字揃えは、デフォルトで「上 (左) 揃え」になっている。このままやると、括弧と句読点など約物と呼ばれるキャラクタが連続した時に自動的に詰められるため、行末がガタガタになる。そこで行の両端を揃えるために文字揃えを「ジャスティファイ」にする。すると、テキストは行末まで引っ張られるので今度は間延びする。とりわけ、画像などをはめ込んだ時、その上または下のテキストが間延びしてしまう、という事態に直面した。

私は書籍など縦組みの組版が比較的多く、文字組みはほとんどがベタ組みだ。特別な XTension を使わずに、できるだけキッチリしたベタ組みができないだろうか。

そしてこれまた行き着いたのは、必要な文字数がピッタリ収まるテキストボックスを作ることだった。20 字ならその 20 字がピッタリ収まるボックスを作れば、考え方としてはキッチリとベタ組みになる。考えてみれば超簡単なこと。行揃えにしてもそうだが、このあたりで私は、QuarkXPress はアバウトにボックスを作ってはダメなんだと考えるようになった。

半年ほど前、仕事でバーの取材をしていた。すっかりはまりこんでしまった。酒にはなく、バーという空間にある。そして「バーとは何ぞや」などと、柄にもなく哲学的なことを追究したものだ。そんな時にあったのが、成田一徹著「二徹の酒場だより」という本だった。バーをモチーフにして、著者が「あまから手帖」誌に連載していた切り絵と文を 1 冊にまとめたものだ。それまでバーに縁がなかった僕は、なかつた僕は、洋酒が並んでいる棚のことを「バックバー」と呼ぶことや、「バーテンダー」とはバー（酒場）とテンダー（見張り番）が合成された言葉だということを、この本で知った。テン

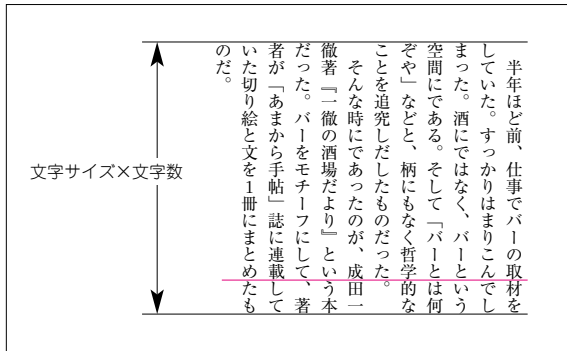
アバウトに本文を切り壊したら、その下の文字が間延びしてしまってみっともない。

半年ほど前、仕事でバーの取材をしていた。すっかりはまりこんでしまった。酒にはなく、バーという空間にある。そして「バーとは何ぞや」などと、柄にもなく哲学的なことを追究したものだ。そんな時にあったのが、成田一徹著「二徹の酒場だより」という本だった。バーをモチーフにして、著者が「あまから手帖」誌に連載していた切り絵と文を 1 冊にまとめたものだ。それまでバーに縁がなかった僕は、洋酒が並んでいる棚のことを「バックバー」と呼ぶことや、「バーテンダー」とはバー（酒場）とテンダー（見張り番）が合成された言葉だということを、この本で知った。テン

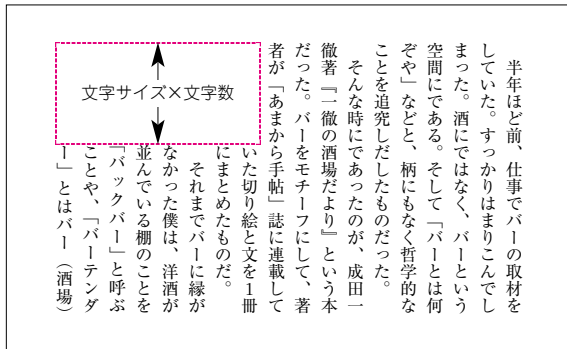
アバウトにテキストボックスを作ったら、全体に文字が間延びした上に、段落最終行では他の行と文字がずれている。段落最終行にはジャスティファイが効かないから揃わないのだが、この行だけが正しい文字送りでもある。

半年ほど前、仕事でバーの取材をしていた。すっかりはまりこんでしまった。酒にはなく、バーという空間にある。そして「バーとは何ぞや」などと、柄にもなく哲学的なことを追究したものだ。そんな時にあったのが、成田一徹著「二徹の酒場だより」という本だった。バーをモチーフにして、著者が「あまから手帖」誌に連載していた切り絵と文を 1 冊にまとめたものだ。それまでバーに縁がなかった僕は、洋酒が並んでいる棚のことを「バックバー」と呼ぶことや、「バーテンダー」とはバー（酒場）とテンダー（見張り番）が合成された言葉だということを、この本で知った。テンダーには「やさしく」という意味もあり、「バーテン

上記サンプルのテキスト全体を、トラッキングを一つずつマイナスして詰めていくと、「-7」でちょうど 1 字詰まった。約物がらみを除けば揃っている。しかし、アバウトで作ったテキストボックスなので、別のボックスでも同じトラッキング値で詰まるとは限らないし、段落書式で「第 1 行インデント」が使えなくなる。



ベタ組の文字揃えのためには「文字サイズ×文字数」のサイズでテキストボックスを作れば基本的にピッタリ揃う。これで揃わないケースは、①連続約物がある、②行頭や行末の約物、③禁則処理がらみ、④欧文や数字を含む場合、の四つにほぼ限られる。これらは調整が必要だが、揃わない原因がわかっているれば対応もしやすい。



ピッタリサイズのテキストボックスであれば、上の図のようにボックスを重ねることで、切り壊しがあってもかなり簡単に対応できる。もちろん約物がらみは調整が必要だ。

文字揃えのための3点セット

文字揃えのためのポイントは3点セットで、①テキストボックスを計算して作る、②「テキストとの間隔」を「0」に、③重ねるボックスの「回り込み」は「0」にする、となる。基本的には行揃えと同じだ。

必要な文字数がピッタリ収まるテキストボックスは次の算式で計算できる。

テキストボックスの高さ(長さ) = 文字サイズ × 文字数

こうして作るテキストボックスがよいのは、切り壊しへの対応が楽でかつ正確にできることだ。たとえば20字のテキストボックスに10字分の枠空けをして画像をはめ込むなら、まず10字分のボックスをテキストボックスの端に合わせて重ねればよい。その上に画像などを張り込むわけだ。これで、画像の上なり下にくるテキストもキッチリ揃って間延びすることはない。

むろん連続約物、行頭行末の約物、禁則処理、欧文を含む場合など文字揃えが崩れるのが避けられない場合はある。そういう場合のみトラッキングで対応したり、H&Jの設定で禁則処理を追い込みにしたり(別の機会に紹介)という処理をしている。

もっともこれはベタ組み専用の方法だ。プロポーショナル組みをする場合は、「ピッタリ」ということ自体が問題にならないからだ。あしからず。

なんで収まらへんねん？

さて、お楽しみはこれからだ。

行揃えの場合も文字揃えの場合も、テキストボックスを計算して作ると書いた。ところが紹介した式に当てはめて計算してボックスを作っても、実際には収まる場合と収まらない場合がある。これは、文字の単位に級数を使っていて、ボックスサイズをミリで計算する場合に起こり、ポイントでは起こらないようだ。

「そりゃそうや。QuarkXPressはポイントで計算しているから、換算した端数の処理の関係で誤差が出るんや」と言われそう。私もそう思っていた。

これはひとまず置いておいて、文字の単位に級数を使いボックスはミリで計算する私は当初、ボックスに壁でもあるのかと思い、収まらない場合に0.01mmをプラスし、その後0.001mmをプラスして回避するようになった。もちろん見た目にはわからない。

しかしこれでは、ボックスのサイズにかなり端数が出るから、作業はけっこううっとうしい。ところが今年になって、Web上の情報で、0.0001mmのプラスでも回避できることを知った。しかもこうすれば、小数点以下第4位は五捨六入(QuarkXPressは四捨五入ではないようだ)されるので、ボックスのサイズはきれいな値になる。

こいつはありがたい、と私はウハウハ喜んで、再利用するドキュメントも順次直しているところだ。

結局、誤差は1万分の1ミリ以下なのだ。そのことがわかれば、級数を使ってもピッタリボックスをストレスなく作ることができる。明暗は、このわずかな値にあったのだ。

計算はQuarkXPressにさせるべし

しかし、この誤差はなぜ起こるのか、という疑問がわいてきた。本当にポイント換算のためだろうか。実は98年の5～6月頃、これにすっかりハマってしまった。

というのは、たとえば12級(3mm)10字、暗算なり計算機で計算し、メジャーパレットにその結果30mmを指定してテキストボックスを作る。これは収まる。ところが13級(3.25mm)10字、同じように32.5mmのボックスを作ると、これは収まらない。

このやり方で実験したところ、10～12級は収まる、13～15級は収まらない、16～18級は収まる……、と3級刻みで交互に収まる場合と収まらない場合が現れた。

級数といっても1Q = 0.25mmだから同じミリ。ボックスも文字もミリで計算しているのだから、収まったり収まらなかったりの原因がポイント換算では説明できないのではないか。

しかも、収まらなかった文字サイズ、たとえば13級の場合でも、メジャーパレットで3.25×10とQuarkXPressに計算をさせてテキストボックスを作れば、いずれの級数の場合も収まることが確認できた。こうなるとますますポイント換算が原因ではないように思える。

しかし私は、コンピュータやプログラムにはとんとんとい上に文系人間なので、ハマッタけれどもいまだにその原因はわからないままだ。読者で詳しい方があれば、ぜひ教えてほしい。

たとえば、QuarkXPressの一般環境設定には「ポイント/インチ」「シセロ/cm」の項があり、それぞれ「72」「2.1967」とある。デフォルトでの1インチ = 72ポイント、1cm = 2.1967シセロとの定義だ。では1ポイントは何ミリかとなると、ハッキリとはわからない。メジャーパレットで計算させると、どうやら0.35277777……とな

る割り切れない数値のようではある。

ようするに換算基準値が関係式で規定されている。文字サイズはQuarkXPressが計算しているわけだからテキストボックスも同様にすべきで、いきなりミリで結果の数値を入力したのでは誤差が出ることもありうるということなのだろうか。

それとも、1文字ずつの文字サイズの計算で、端数処理によるごくわずかな誤差で指定の級数より大きいものと小さいものができるのだろうか。10個のボールを入れる箱を作るのに、10個が入るはずのサイズを計算して作るのと、1個ずつのボールを実測して作るのでは、誤差が出ることはあるだろう。これだと収まる、収まらないを説明するには有効なように思われた。

しかしコンピュータは、端数処理といってもそれは表示のために、計算の結果としてはフル数値を持っているのではなかったか。それに、10字でも20字でもやはり誤差は1万分の1ミリ以下なのだ。そうなる上記もあてはまらないように思える。エクセルでシュミレーションしても、3級さざみという状況を確認することはできなかった。

というわけで大いに疑問だが、これにハマってはい仕事にならない。しかしハッキリしていることが一つだけある。QuarkXPressに計算させればピッタリ収まるという事実だ。もしそれでも収まらなかったら、再計算させればよい(3回目で収まったこともある)。前半で紹介した条件をクリアしていれば、行にしても文字にしてもまず確実に収まることを日々経験している。だから計算するならQuarkXPressで行う。これがベターだと思う。

もちろんこれは、文字とそれを収めるボックスという二つの計算が絡むテキストボックスだけに言えることが、1万分の1ミリ以下で誤差が出ることもあるということを知っていると、役に立つことがよくある。

なお、ボックスのわずかな誤差にそんなにこだわらなくても、H&Jの設定でこの程度の誤差は吸収できる、とWeb上で教えていただいた。H&J設定の「ジャスティファイ設定」でかなや漢字の「最少」値を「-10～-20」程度にする。この程度なら文字揃えにも影響せずピッタリ収めることができる。1行文字数が少ないと効かないこともあるが、多い時は有効だ。参考までに。

第2回

マスターページとプラトニック

マスターページアイテムには指を触れるな

ある仕事で「慕情」という言葉についての講演を聞く機会があった。もともとこの言葉は日本語になかったが、映画「慕情」のヒットによって辞書にも登場するようになったという。もうずいぶん昔の映画だ。

広辞苑によると「慕情」は、「したわしく思う心」とされている（新村出編『広辞苑』第四版、岩波書店 1991年）。俗に言えば「愛しいが、相手に指一本触れないこと」だと講演者は語り、その典型を映画「男はつらいよ」の寅さんと言ったものだった。

このところすっかり聞かないが、「プラトニックラブ」という言葉も「指一本触れない」という意味ではほとんど同意だろう。これまた広辞苑によると「プラトニック」は「精神的な」とされている。

実は、今回は QuarkXPress のマスターページをテーマにしたいのだが、いったい「慕情」や「プラトニック」とマスターページにどういう関係があるのか。ちょっとお楽しみいただきたい。

なお、今回も基本的内容なのでバージョン 3.3 でも 4.0 でも同じことだが、サンプルは 3.3 を使用した。

オーバーフローマークの嵐が……

ある時ガイドブックの仕事をした。A5 判変形縦長のサイズに、イベントやホテルなどの施設が紹介されているもので、定型的なパターンに写真やテキストを埋め込んでいくスタイルだった。

私はまず、基本的なデザインの指示に基づいて一つのページをドキュメント上でレイアウトし、それをそのま

まマスターページにコピーした。もちろん、テキストはページごとに入れ替わるので、マスターページ上のテキストボックスは空にする。このマスターを使ってせっせと作業を終え、初校、再校と仕事は進んだ。

どの段階だったか忘れたが、ページの入れ替えがあった。これ自体はよくある話だ。特に造作はない。

ところがその後の作業で、どうもモニタ上の雰囲気が違う。よく見るとそれらはテキストボックスのオーバーフローマークの嵐なのだ。テキストがボックスから溢れているわけではないのに、ことごとくテキストボックスにそれがある。なぜ？

よく見ると、もともと配置されたテキストボックスの裏側にもう一つのテキストボックスが出現している。たまたまテキストボックスのバックグラウンドを「なし」にし、回り込みを「アイテム」にしていたので、その下にできたテキストボックスには何もテキストが入ってなくてもオーバーフローとされ、そのマークが見えたというわけだ。

下のボックスは空なので、おそらく出力に問題ないとは思われたが、余計なデータを含むことになるし、作業上も目障りで仕方がない。一括して削除するにはどうすればよいかもわからず、やむをえず手作業で削除したのだったが、これでは再び同じことが起こる可能性が高い。

よく考えたら、それまでも同じようなことが何度かあった。とりわけ、マスターページ上に空のテキストボックスを複数配置すると、ページが増える時に本来のテキストボックスではないボックスにチェーンされたりしてとまどったものだった。

このようなことのない快適なマスターページの使い方はどうすればいいのだろうか。それにはまず、マスターページのメリットから考えてみたい。

マスターページの二つのメリット

マスターページと実際に編集作業を行うドキュメントページとの関係は、いわば親亀と子亀のようなものだろう。私はマスターページに、最大限に生かしたいメリットが大別して二つあると思う。

一つは、マスターページで作った定型フォーマットを使って簡単にドキュメント上の各ページを編集できることだ。QaurkXPressのマスターページは一つのドキュメントで最大128種類作ることができる。実際にこれだけ作ったら管理が大変だが、複数のマスターページを使用するのは、なかなか使い勝手がよい。

もう一つは、マスターページで加えた変更が、そのマスターを適用しているドキュメントページにそのまま反映することだ。何十ページにもわたって同じ変更がある場合など、これは実にありがたい。

ところでQaurkXPressでは、環境設定の「一般」に「マスターページアイテム」という項目があり「変更箇所保持」と「変更箇所消去」が選べるようになっている。

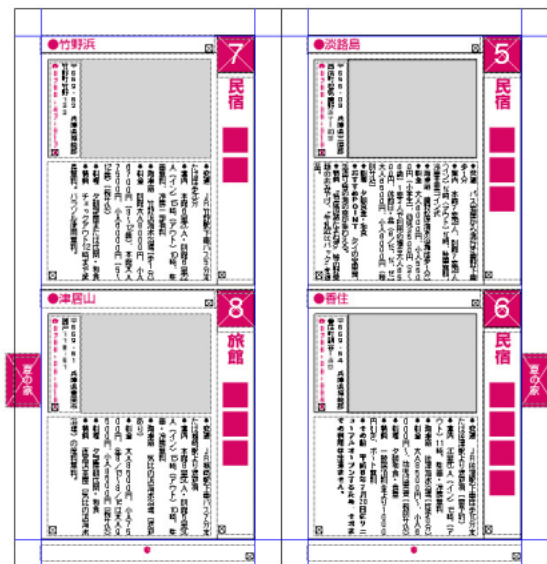
「変更箇所保持」は、ドキュメント上でマスターページアイテムに変更を加えた場合、マスターページが再適用されてもその変更が生きる、というものだ。基本的にはこれを使うのが賢明だろう。「変更箇所消去」には少し不安を感じる。

その状態で、なおかつマスターページからドキュメントページへの反映の風通しをよくするためには、ドキュメント上で、できるだけマスターページアイテムを変更しないことではないかと思う。

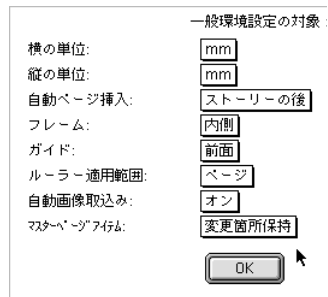
テキストボックスと「変更」の関係

では、QaurkXPressはどういう場合に「変更」とみなすのだろうか。もちろんアイテムのサイズや位置を変更すれば、これらは一目瞭然だ。

ところがテキストボックスの場合は、サイズや位置を



無惨にオーバーフローマークが出現したドキュメント。テキストボックスが重なって、下のボックスがオーバーフローした。なお、グレーの四角や■は写真や固有名詞をカットしたものだ。



環境設定→一般の設定画面。多くの場合、マスターページアイテムは「変更箇所保持」だろう。

変更しなくても、テキストを入力しただけで変更となる。考えてみればもっともなことだが、これに気がつくまでに時間がかかった。それは「自動テキストボックス」というものがあるからだ。

これは、テキストがオーバーフローした時に自動的に新たなページとテキストボックスを作り、そこにオーバーフロー分を流し込んでいくもの。もちろん、複数のテキストボックスを連結することも可能で、連結されたボックスはすべて自動テキストボックスとして扱われる。

自動テキストボックスにはマスターページ上でテキストを入力することができないが、ドキュメント上でいく

新規

USレター
 その他
 364 mm

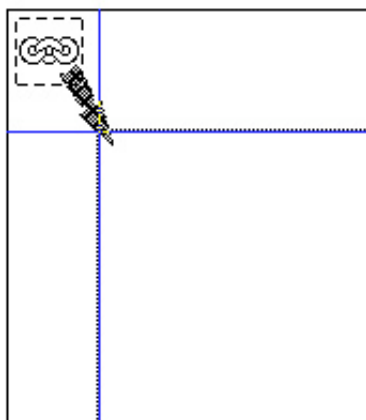
コラムガイド
 コラム数: 1
 コラム間隔: 4 mm

デフォルトの組み方
 横組み 縦組み

テキストボックスの自動作成

15 mm
 15 mm

新規ドキュメントの作成時に「テキストボックスの自動作成」をチェックすると、自動テキストボックスが作成される。



マスターページ上で見た自動テキストボックス。連結アイコンがつながっていることで確認できる。連結ツールを使って、マスターページ上で自動テキストボックスのオン、オフをすることも可能だし、複数のテキストボックスをつなぐこともできる。

らテキストを入力しても「変更」とはみなされず、マスターページ上でサイズなどを変更すれば、それがドキュメントに反映される。

私は、すべてのテキストボックスがこれと同じだと思っていたのだ。しかし、自動テキストボックス以外の通常のテキストボックスはそうではなかった。

つまり、マスターページ上で空のテキストボックスを作り、ドキュメント上でテキストを入力した場合、そこにマスターページが再適用されると、そのテキストボックスは「変更」とみなされて、下に空のテキストボックスが再出現するということになるのだ。

では、どんな時にマスターページが再適用されるのか。

これは主に二つの場合で、一つは手動で再適用する時、もう一つは特に見開きマスターを使用している場合でのページ移動だ。移動によって左右が入れ替わると、以後の全ページにマスターが再適用される。これは、時として予期しない再適用となるケースが多い。できるだけ避けたいところだ。

最初に紹介したケースは、マスターページの空のテキストボックスとページ移動という、まさにこの二つが重なったのだった。

ならばマスターページには、自動テキストボックス以外の空のテキストボックスを作らなければよいではないか。私はこう考えた。

はじめからマスターページになければ、予期しない再適用があっても、ドキュメント上に再出現することはない。それに、空のテキストボックスを作ってもマスターページでの変更が反映できないのなら、結局メリットは半減する。それならむしろドキュメント上でパターンを作ってライブラリに登録し、それを使っても同じではないだろうか。

マスターアイテムは必要最小限に

マスターページアイテムと「変更」の仕組みがわかってくると、私のマスターページの作り方は変わった。

それは、ドキュメント上で変更を加える予定のアイテムをマスターページ上には作らず、必要最小限にとどめることだ。同時に、ドキュメント上ではマスターページアイテムを決して触らない。これで確実にマスターページでの変更がドキュメントページに反映されるし、誤って触っても、冒頭のようなケースは最小限になる。

するとマスターページ上に作るアイテムはどうなるか。

たとえば書籍だと、本文の入る自動テキストボックス、ノンブル、柱、その他ページをあしらうアイテムとなる。多段組のページものでは、本文用のテキストボックスも作らずにマスターガイドだけにとどめている。テキストボックスはドキュメント上でガイドに沿って作るのだ。

パターンもの場合も、ノンブルや柱、その他定型なアイテムだけにし、各ページで異なるテキストの入る

ボックスは作らないようにしている。これらはライブラリを利用したり、コピー&ペーストで対応している。

もちろん、同じパターンで柱だけが違うとなどというケースはよくある。こうした場合はマスターページをコピーして、柱だけ入れ替えるわけだ。つまり柱の数だけマスターページを作ることになる。

ようするに、複数のページで全く同じアイテムを同じ位置で使うなら、それがマスターページで作るべきアイテムと考えていだろう。当然だが、マスターページの数が増える。全体のデザインが変更になればそれらを変更しなければならない。それだけに、最初によく検討を重ねて作る必要があるが、各ページをいちいち変更するよりは、はるかに手間が省けることも確かだ。

最小限アイテムにしたなら、マスターページからドキュメントページへの反映といっても効果減るのではないと思われるかもしれない。しかし、たとえば右のサンプルのようなケースでもストレスなく変更できるので、余計な作業は減っても、効果が減るわけではないと思う。

マスターガイドを生かそう

マスターページアイテムを最小限にとどめると、フォーマットをページ編集に生かすというもう一つのメリットとの関係はどうなるか。

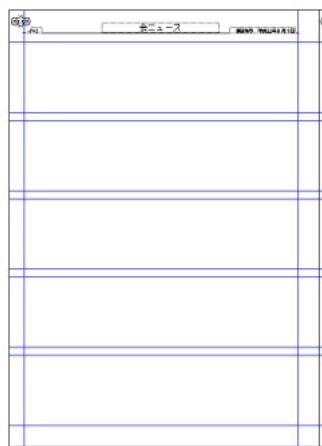
私は、マスターガイドを活用することがミソだと思う。

マスターガイドとは、マスターページの「ページメニュー」→「マスターガイド」で作るマージンガイドとコラムガイドのことで、デフォルトだとブルーのラインで表示される。新規ドキュメント作成の際にも同様の設定をするが、これが最初に作られるマスターガイドだ。

作業感覚的には、マスターページ上に作る自動テキストボックスのサイズまたは版面と考えてもいいだろう。

そしてポイントは、このマスターガイドを連載第1回で紹介したピッタリサイズのテキストボックスに合わせることだ。これが文字揃えや行揃えのベースとなる。

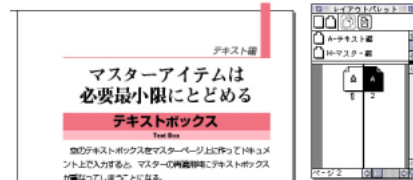
このガイドがあれば、自動テキストボックスを作らない場合でも、ドキュメント上でマスターガイドに沿って、マウス操作だけでピッタリサイズのテキストボックスを簡単に作ることができる。サイズを変えるなら、それを



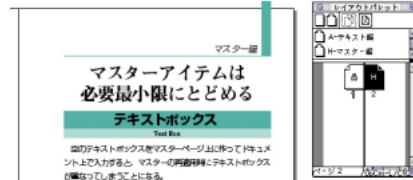
アイテムを最小限に絞った縦組み多段組み用のマスターページ。
段組みではテキストがさまさまに入り組む編集になり、一定の自動テキストボックスでは対応できない。よって柱、ノンブル、ラインなどの定型アイテム以外は、マスターガイドだけにしている。
もちろん、文字組みスタイルが一定している書籍などでは自動テキストボックスを作っておく。

モノクロなのでわかりにくいですが、カラーや柱の違う別のマスターを適用した例。内容は同じだがカテゴリーが変更になるというケースだ。作業はワンクリックで終わる。このマスターはカラーアイテムと柱、ノンブルだけの空のテキストボックスがあれば確実に重なるケースで、表面うすくできても重なったボックスが見えていないだけ、ということも少なくない。

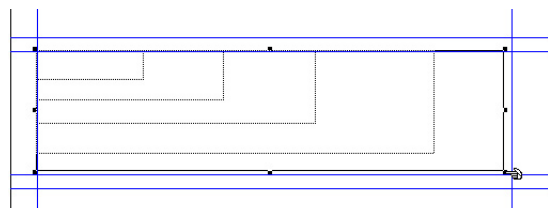
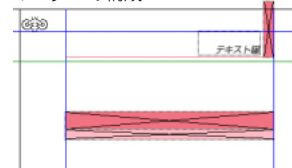
A マスターを使用



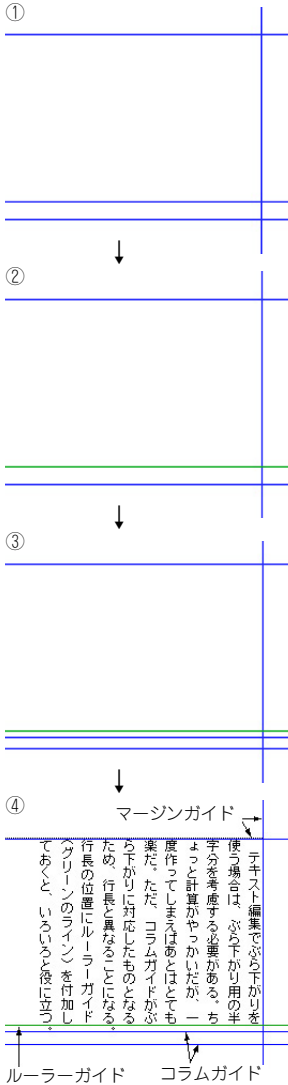
H マスターに変更



マスターの構成



ピッタリサイズのテキストボックスに合わせたマスターガイドがあれば、ドキュメント上でそれに沿って、正確なボックスをマウス操作だけで簡単に作成できる。
そのボックスをベースに、文字サイズや行送り値を増減すれば、行揃えや文字揃えを崩さずにリサイズすることも容易だ。



ぶら下がり用の多段組マスターガイドの簡単な作り方

(例)
14 級(3.5mm)14 字 5 段組段間 5mm、用紙 A4 縦

- ①まず、ぶら下がりのない状態のマスターガイドを作る。テキストボックスのサイズは $3.5 \times 14 \times 5 + (5 \times 4) = 265\text{mm}$ マージン値はドキュメントサイズとの差を天地に割り振って算出する。ここでは天を 17mm、地は 15mm とした。
- ②各段の行末の位置にルーラーガイドを引いておく。
- ③マージンガイドの地の数値とコラムガイドの数値から、半文字分をマイナスする。0.0001mm 少なくするのがミソ。

マージンガイド			
天:	17 mm	内側:	
地:	15-1.7499	外側:	

コラムガイド	
コラム数:	5
コラム間隔:	5-1.7499

するとルーラーガイドが行末を示し、コラムガイドがボックス(コラム)間隔を示す。

- ④ドキュメント上にボックスを作ってテキストを入れた状態。行末を示すルーラーガイドがあると、何かと便利。

ベースに文字サイズや行送り値を増減すれば OK だ。もちろん、ライブラリなどからアイテムをもってくる場合の目印にもなる。

テキスト編集でぶら下がりを使う場合は、ぶら下がり用の半字分を考慮する必要がある。ちょっと計算がやっかいだが、一度作ってしまえばあとはとても楽だ。ただ、コラムガイドがぶら下がりに対応したものとなるため、行長と異なることになる。行長の位置にルーラーガイド(グリーンのライン)を付加しておく、いろいろと役に立つ。

マスターガイドだけでは作れないサブのテキストボックスが必要な場合は、一度マスターページ上でテキストボックスを作り、その位置にルーラーガイドを引いて、ボックスは削除するのがよいと思う。

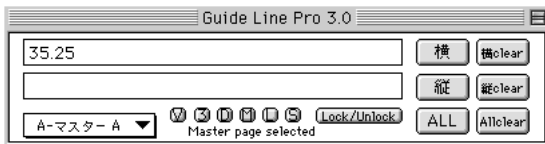
ただ、残念ながらデフォルトでは正確なルーラーガイドが引けない場合があるので、「Guide Line Pro」などの XTension がおすすめだ。

ところで、テキストボックスがピッタリだと、特に縦組みで 1 行目のルビや傍点などに困る。これは次ページのサンプルのように「アイテムメニュー」→「設定」にある「第 1 行目のベースライン」の設定で対応可能だ。

縦組みの場合は、マスターページでテキストボックスのサイズだけ(マスターガイドはそのまま)変更して「第 1 行目のベースライン」を設定しおけば、ドキュメント上で触る必要もない。

横組みでベースライングリッドを使うならさらに簡単に、マスターページのテキストボックスの上下を、ルビや傍点を吸収しうる任意のサイズ分マスターガイドよりも大きくしておくだけで OK だ。

なお、ものによってはマスターガイドと別にマージンサイズを設定したい場合もあるし、逆に本来のマージンガイドとして使用するのがよい場合もある。これらはケースバイケースで、要はドキュメント上で作業がしやすいように各種ガイドを正確に引いておくことだろう。



Guide Line Pro 3.0
マスターページも含め、正確なルーラーガイドが複数同時に引ける。
Apple Script で動くシェアウェア(1,500 円)。
インターネットで入手可能 (AppleScript Irregulars)
URL: <http://www.dd.iij4u.or.jp/~xyz/>

あなたは外から？ それとも中から？

マスターガイドを作る場合、マージンから考えるのか、ボックスサイズから考えるのかという問題がある。つま

り外から考えるか中から考えるかだが、QuarkXPressの仕様はマージンを設定するものだし、組版指定でもマージンを指定されることが多い。

おそらく私の仕事が文字組み中心だからだろうが、私はほとんどの場合、中から考えている。指定の文字数、行数、段数などが収まるテキストボックスのサイズをまず計算し、それとドキュメントサイズの差をマージンに振り分けるわけだ。

結果的に指定マージンと誤差が出ることもある。それでも私はテキストボックスのサイズを優先し、マージンは指定の近似値にする。ごくわずかの差なので見た目にはわからないし、実際問題、製本段階の裁断でわずかな狂いは出るわけだ。少なくともテキストものでは、文字組みの正確さや作業のしやすさを優先する方がベターだと思っている。

マージンの設定はボックス作成と逆の作業になるので、そのサイズを前回書いた「QuarkXPressに計算させる」ことはできない。そこで「1万分の1ミリ」が活躍する。マスターガイドに天地左右の数値を入力し、天地、左右のそれぞれの一方から0.0001mmをマイナスするのだ。

単位に級数を使っていれば、マスターガイドもテキストボックスも必ず0.25mm単位のきれいな数値になるので、いろんな作業がやりやすくなる。

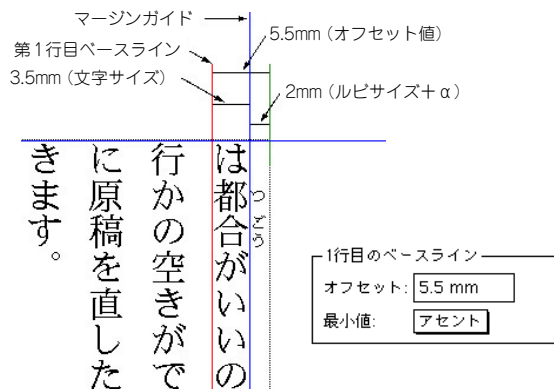
* * * * *

おおむね以上のようなマスターページの作り方をすると、ドキュメント上でマスターアイテムに変更を加える必要がなくなってくる。当然、マスターページでの変更はドキュメントページに反映し、校正時での変更もストレス少なく対応することができるというわけだ。

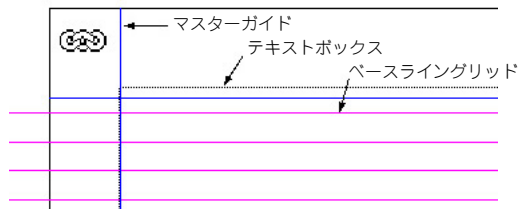
これぞまさしくマスターページとの「指一本触れない」関係、「慕情」とも「プラトニック」とも相通じるような関係だと思うのだが、いかがだっただろうか。

ただし、見開きの左右で異なるマスターページを使用しているケースなど、マスターページでの変更が直ちにドキュメントページに反映しない場合がある。手動での再適用が必要になるが、こうしたイレギュラーも含まれていることを付け加えておきたい。

●えでいっとはうすのホームページ別館 QuarkXPress 奮戦記
URL: <http://ha3.sekyou.ne.jp/home/ogn/>



縦組みでの第1行目のベースラインの設定例。サンプルは文字サイズ14級(3.5mm)とした場合。コツは、もとのサイズより右側にはみ出したテキストボックスにすること。ルビと被ルビ文字の間に微妙なスペースがあるので、はみ出す幅はルビサイズ分だけでは足りない。しかし大は小を兼ねるので、ようするにルビが収まるサイズであれば任意の数値でかまわない。これはみ出し幅と文字サイズを足した数値を「オフセット」値として入力すると、上のような関係になる。この場合、文字揃えに「ベースライン」を使うとうまくいかないのが要注意。「センター揃え」がいいだろう。



横組みでベースライングリッドを使用する場合は、ルビが収まるサイズ以上の任意幅だけ、テキストボックスを上にはみ出しておく。これだけで1行目のルビや傍点に対応できる。早い話が、行揃えはグリッドに任せるので、テキストボックスはアバウトでかまわないわけだ。ベースライングリッドを使わない場合は、縦組みの場合と同じく「第1行目のベースライン」をオフセットする。

マスターガイド			
マージンガイド			
天:	20 mm	内側:	15-0.0001
地:	17-0.0001	外側:	15 mm
コラムガイド			
コラム数:	2		
コラム間隔:	10 mm		
OK		キャンセル	

マスターガイドの設定では、天地、左右(内側、外側)のそれぞれで、そのいずれかから0.0001mmマイナスするのがコツ。上の例は16.9999mm、14.9999mmが実際の数値だが、五捨六入されてそれぞれ17mm、15mmと表示される。これによって作られるテキストボックスも実際には0.0001mm大きい、同じく五捨六入できれいな数値になり、文字数や行数がピッタリ収まる。

第3回

自由と不自由は紙一重？

マニュアルな表組み作業を考える

QuarkXPressの表組みはどうんざりする作業はない、というのがほとんどのクォーキアンに共通する感覚ではないだろうか。

ただでさえドキュメントの制作方法は人によってさまざまだが、表組みとなるとそれに輪をかけてさまざまになるようで、他人が作成した表組みデータはさわりたくないという声も聞く。さもありませんというところだ。

表組み用のXTensionは少なくないが、私は表組みに限らず、標準以外のXTensionを使うことははなはだ少ない。XTensionといえども基本的にはQuarkXPressの標準機能を利用しているものが多いから、仕組みがわかれば手動で対応できるものもある。だから、表組みもライントoolとボックスを駆使して地道にやっている。

考えてみれば、表組みXTensionの紹介はあちこちにあるが、それを使わない方法の紹介は少ないように思う。そこで今回は、手作業での表組みについて考えてみたい。

ひと口に表といってもいろいろなものがあるが、ここでは枠線のある表を対象にし、表計算に準じて「セル」「行」「列」という用語で話を進めよう。

QuarkXPressの作表はなぜ面倒？

まず、QuarkXPressの表組み作業はなぜ面倒なのか。私は当初、表の各セルの中央に正確にテキストを置くにはどうしたらいいのか、たったこれだけのことがわからなかった。

テキストボックスを作って数行の表のテキストを入力。横の位置はタブで調整して、それからラインを付加する。

しかしライン位置は目測だから、横線も縦線もなかなかうまくいかない。逆に、先にラインで表を作ってセルの中央にテキストを配置するのも、テキストボックスの位置を目測であれこれ調整するが、プリントすれば明らかにズレているということが少なくなかった。だから、こんなことを繰り返すのはたまらない、と思ったものだ。

しかし、ラインとボックスを別々に作ってアバウトに配置すればずれるのは当たり前だ。こうしたことから結局、面倒に感じる原因は次の2点ではないかと考えるようになった。

- ①一本ずつのラインで作表しなければならない。
- ②ラインとボックスを別々に作らなければならないので、テキストの位置調整が難しい。

ならば、常にラインとボックスの関係を意識し、一定の原則を設ければよいのではなかろうか。ずいぶん時間がかかったが、これに気づくと、少なくとも位置調整に悩まない比較的楽な方法がいろいろ見えてきた。


えでいっとはうす流の表組み四原則

そこで、実際の作業で私が表組みに際して意識している原則、いわばえでいっとはうす流の作表原則を紹介すると、おおむね次の4点だ。

①ミリ単位程度のできるだけ単純なサイズの表を作る

あらかじめ表全体のサイズを決めて作業に入ると作業はずいぶん楽で、そのサイズも5mmとか10mmなどミリ単位のできるだけ単純なサイズにするのが作りやすい。

②表枠は全てラインツールで作る

私は表枠の作成に全てラインツール（ここでは直角ラインツール 、以下同）を使用している。これには、他にも次のような方法がある。

- ・外枠にはボックスのフレームを使用し、中をラインツールで作る。
- ・外枠にはボックスのフレームを使用し、中の横線を「段落罫線」で、縦線をラインツールで作る。

それぞれに一長一短があるが、私が全てラインツールで作表するのは、次のような理由による。

- ・フレームは表示モードによって見えないことがある。
- ・ラインツール一つでたいがいの表が作れる。
- ・「複数アイテムの配置」を活用できる。実はこれが最大の理由で、ボックスのフレームでは使えない。

③テキストボックスはできるだけ表やセルと同サイズに

ラインとテキストの関係を考えると、表またはセルとテキストボックスのサイズを同じにしようとするのが、位置調整が一番楽だと思う。もちろん小さなボックスもOKだが、その場合は「複数アイテムの配置」の「均等配置」を活用して揃えるようにしている。

④セルの高さとテキストの行送りを同じにする

表の複数行にまたがるテキストボックスでは、セルの高さとテキストの行送りを同じにする。このためにもセルを単純数値にするのがよい。私は日頃から文字単位に級数を使っているが、こうするとさらに楽。セルの高さが10mmなら行送りは40Q（10mm）と、関係が簡単だ。

表組みに適した三つの環境

表組みに関する私の環境についても紹介しておこう。

まず一つは、テキストボックスの「テキストとの間隔」を「0」にする。デフォルトでここは「1pt」になっているが、そのままだと狭いボックスで差し支えがでる。

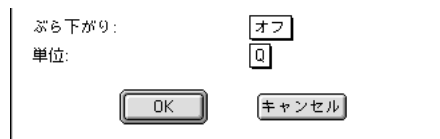
それから「回り込み」の設定を「なし」にする。これもデフォルトは「1pt」になっているが、表組みではボックスが接することがよくあるので差し支えがある。



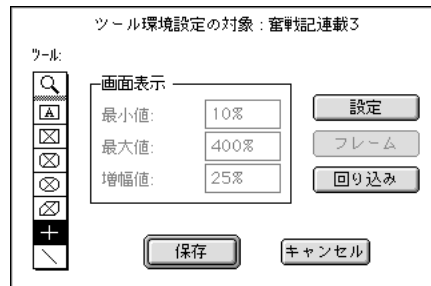
アイテムの「設定」で左下の「テキストとの間隔」を「0」にする。右上の「ボックス内の行位置」を「センタリング」にするのにも注目しておいてほしい。



表組みではボックスが接することが多いので、「回り込み」は必ず「なし」にしておくのが重要。



日本語環境設定で文字の単位に「Q」を使うと、単位系をミリに統一できるのでメリットが大きいと思う。ちなみに、これまでの連載で述べたピッタリサイズのテキストボックスやそれに合わせたマスターガイドが引いてあれば、いっそう作りやすい。たとえばこの紙面の本文コラムは12級（3mm）25字だから幅は75mmとなり、コラム幅に合わせた表を作る場合も作業が楽。もちろんサイズによっては0.25mm単位で端数はでるが、ランダムな数値よりはるかに扱いやすい。無理にすすめるはしないが、日頃ポイントで作業している方も、作表の時だけ切り替えてみるという手はある。



ツール設定画面。ツールをダブルクリックすれば開く。あらかじめ設定しておけばいちいち設定しなくてもよいから楽だ。

ラインの設定

種類:		モード:	左端
先端の形状:		左端:	
太さ:	▶ 0.4Q	横座標:	
カラー:	■ ブラック	縦座標:	
シェード:	▶ 100%	横座標:	

ラインの基本サイズも設定しておくのが楽。直線を引くには直角ラインツールがよい。垂直、水平に固定されているので、通常の表枠なら斜線マークのラインツールより扱いやすい。

直角ラインツール ラインツール

項目	データ
A	50,000
B	5,000

●作表サンプル。セルの高さは10×30mm、3行2列なので全体サイズは30×60mm。ごく簡単な表だが、これでもテキストボックスの作り方は少なくとも5通り以上ある。

方法① 一つ一つのセルにテキストボックスをつくる

項目	データ
A	50,000
B	5,000

各セルにテキストボックスをつくる方法は、セルサイズがまちまちの場合などに有効だ。なおサンプルのデータ欄の数値は「5,000」のセンタリング位置を基準に「右インデント」を設定した上でテキストを「右揃え」にした。

項目	データ
A	50,000
B	5,000

ボックスは原則としてデータの入る方向に連結する。それぞれ独立したままだと、校正時に手間がかかって仕方がない。しかし、場合によっては上の例のように項目部分を縦に連結するなど、組み合わせを工夫するのも有効だ。このあたりは手作業でつくるメリット。

そしてもう一つは「日本語環境設定」を「Q」にする。すると文字も行送りもラインの太さも「Q」で表示される。1級は0.25mmなので事実上ミリと同じだから、単位をミリに統一できるというメリットが生まれる。

これらは、作業の前にあらかじめツール環境設定で設定・保存しておけば、あとでいちいち設定する必要はない。また、ラインの太さも同様にしておくとうい。以下、この設定を前提にする。

表組みテキストボックスの五つの方法

さて左の作表サンプルを参考に考えてみよう。各セルを10×30mmとしたので3行2列のこの表は全体で30×60mm、ラインの太さは0.1mm(0.4Q)とした。

こうして方針を決めておけば、表枠の作成は簡単だ。太さ0.4Q長さ60mmの横線を任意の位置に引き、10mm間隔で垂直方向に3本コピー。次に最上部左端の座標位置に合わせて同じ太さで長さ30mmの縦線を引き、右へ水平に30mm間隔で2本コピーすればOKだ。

単純な表だが、実はこれには少なくとも五つ以上のテキストボックス作成方法がある。

①一つ一つのセルにテキストボックスをつくる

まず第一は、セルの一つ一つの中にテキストボックスを作る方法だ。実はこれ、基本的に表組みのXTension「テーブルワークスプラス」の方法。XTensionは使わなくても、そのエッセンスは活用させていただこう。

表枠を作ったら、一つのセルに同じ大きさのテキストボックスを作る。テキストボックスはラインより背面がよい(以下同)。アイテム「設定」の「ボックス内の行位置」とテキストの「文字揃え」を「センタリング」にすることで、テキストはボックス(つまりセル)の中央に配置できる。あとは各セルにコピーすればOK。

このサンプルだけ小さめのボックスをアバウトに作った。この場合は上記に加え、ボックスとそれを囲むラインを選択して「複数アイテムの配置」で「均等配置」にすれば、ボックスをセルの中央に配置できる。あとは各セルにコピーすればよいが、先にコピーをし、まとめて「均等配置」にしてもOKだ。

この方法がいいのは、表もテキストボックスも、比較的アバウトに作るができること。また、ボックスが独立しているから、あるセルは1行、別のセルは2行などという場合でも簡単に対応できる。さらに、セルごとにリターンやエンターでテキストを区切れれば、それぞれに違うスタイルが適用できる。文字単位のスタイルが使えない3.3J以前では、表面上は同じ行の中に複数のスタイルを適用する方法として、表以外でも使える。

これらのメリットは、セルのサイズがそれぞれ違う表で特に有効だ。

デメリットは、ボックスだらけになること。グループ化しておかないと、移動の際に忘れ物をする可能性もあるし、ボックスや連結が多い分データ量も大きくなる。

②セルを縦抜きにしたテキストボックスをつくる

第二は、セルを縦抜きにしたテキストボックスを作る方法。セルの高さがすべて同じなら、この方法が楽だ。

テキストボックスは三つのセルと同位置同サイズにし、セルの高さとテキストの行送り値を同じにする。そして「ボックス内の行位置」を「センタリング」にすれば、セル内のテキストの上下位置は全て確実にセンターにくる。ライン間隔とテキスト間隔が同じなのだから当然だし、覚えておいて損はない方法だと思う。

この方法では、行列を入れ替えてテキスト入力することになるので、うっかりすると行ズレを起こしかねない。十分に注意しよう。

③セルを横抜きにしたテキストボックスをつくる

行単位で、表の横幅と同じサイズのテキストボックスを作るのが第三の方法。行の高さは違うが上から下まで列幅は変化しない、という表に有効だ。

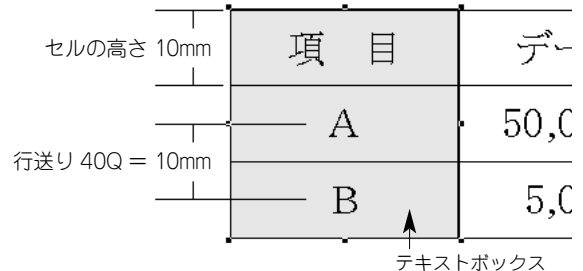
テキストの上下位置は方法①と同様。ボックスはやはり連結しておくことで作業が楽だ。

テキストボックスの高さを文字サイズに合わせたり、テキストが複数行であればピッタリサイズにしたり、あるいは行数によってサイズを決めるなど、一定の原則をもって作成すれば、上下のラインとボックスをすべて選択して「複数アイテムの配置」で「均等配置」することで、セルの高さがランダムな表も簡単に、かつラインと



「複数アイテムの配置」は水平方向と垂直方向を別々に作業する。

方法② セルを縦抜きにしたテキストボックスをつくる

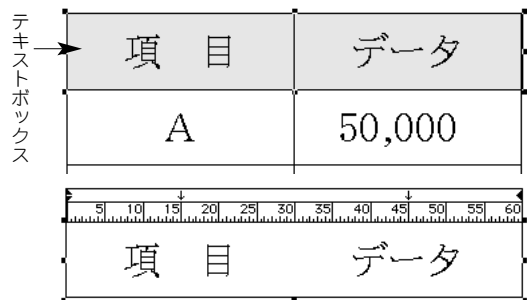


セルの高さが10mmなら級数で40Qだから、テキストの行送り値も40Qにする。ボックス内の行位置が「センタリング」なら、セル内のテキストの上下位置は確実にセンターにくる。空白セルにはリターンを入力しておくことを忘れずに。

項	目
事	例 A
事	例 B

縦抜きボックスは、項目の均等割付にも有効で、トラッキングで対応するより楽だ。ボックス幅をセル幅より少し狭くするが(左図)、段落書式で左右にインデントを設定して「強制割付」する。この場合は最後の行にもリターンを入れる。だから、このオーバーフローマークには意味がある。

方法③ セルを横抜きにしたテキストボックスをつくる



横抜きにしたテキストボックスでは、テキスト揃えにタブを使う。サンプルは15mmと45mmの位置にセンタータブを設定。データ欄の数値部分にはカンマタブを使用したがる、位置は目測だ。もちろん右揃えタブでもOK。

項 目	データ
A (消費税)	50,000 (2,500)
B	5,000

横抜きにしたテキストボックスなら、上のようにセルの高さが違ってても「複数アイテムの配置」で揃えやすい。サンプルはピッチリサイズのテキストボックスを使った。

方法④ 外枠と同じサイズのテキストボックスを作る

項 目	データ
A	50,000
B	5,000

一つのテキストボックスなので最もシンプルな方法。しかし、一部のセルだけ2行などイレギュラーがあると対応しにくい。

方法⑤ 四つの方法を組み合わせる

項 目	データ
A	50,000
B	5,000

左側には方法②を、右側には①を組み合わせた例。テキストボックスの大きさの違いが区別できるだろうか。

項 目	データ1	データ2
A	50,000	3,000
B	5,000	1,000

上記の応用例。左側には方法②、右最上段は③、数値欄は①を使った。「データ」の欄は単純なセンター揃えなのでタブでも簡単だが、数値欄は段落書式の方が楽なのでセル単位のボックスにしてみた。むしろ方法②でもよい。また数値欄は「50,000」のセンタリング位置を規準にし、右インデントを設定して右揃えにしている。

テキストの間隔を揃えることができる。

この方法では横方向のテキスト配置にタブが必要になる。セル幅に応じてタブを設定するが、正確にするには多少の計算が必要で、これがやや手間ではある。その意味でも、セル幅はできるだけ単純な数値にしたいし、文字サイズの倍数にするのもタブ設定には有効だ。

もっともタブには、その設定を変更するだけで全体のテキスト位置を一括訂正できるというメリットもある。

④外枠と同じサイズのテキストボックスをつくる

さらに第四は、外枠と同じサイズのテキストボックスを作る方法。これは方法②と③をミックスしたもので、上下位置は②と、左右の配置は③と同じだ。サンプルのように、セルの高さが一定で、表中のすべてのラインが十文字に交わる表に有効だ。

この方法は、QuarkXPress上で数値などのテキストを直接入力する場合によい。左手でタブ、右手でテンキーを打てばかなりの高速入力が可能だ。

⑤四つの方法を組み合わせる

最後は、これまでの四つの方法をミックスする方法だ。たとえば、均等割付が必要な項目だけは方法②で作って残りは④、全体は方法④で作って一部のみ①あるいは②を用いるなど、状況に応じて使い分けられる。こうして、ボックス作成の方法は5種類以上になる。

どう使い分けるかのポイントは、直しやすさ。校正などで訂正が入るのは日常茶飯事なだけに、実は作りやすさより直しやすさの方が重要だ。

たとえば行や列の削除や追加でその分だけ表のサイズが変わるのなら、どの方法でも比較的簡単に対応できるが、列削除の場合は、タブを使用する方法③と④にはやや厳しい。

表のサイズを変えずに行削除する場合、対応しやすいのは方法①と③だ。①は横方向にグループ化することで、③と同じようにも扱える。列削除なら②だろう。

いずれも削除・追加ののち「複数アイテムの配置」で再「均等配置」するが、この点でも全てラインツールで作っておくメリットがある。また方法①は、連結順序を変えることで行列置換にも対応できる。

表のリサイズ

手作業で作成した表にとってリサイズ、特に全体の拡大・縮小はかなり辛い作業だ。このために Illustrator で作表して貼り込む方法をとっている方も少なくないようだ。しかし、次のような方法もある。

① 4.0J では複数アイテム同時リサイズが可能

バージョン 4.0J では複数のアイテムの同時リサイズが可能になった。グループ化すればドラッグ操作で全体をリサイズ、コマンド+ドラッグならテキストも同時にリサイズできる。また該当する複数のアイテムを選択して、メジャーパレットの「W」「H」欄にそれぞれ数値を入力したり比率を掛けても OK だ。

② EPS ファイルでページ保存して拡大・縮小

バージョン 3.3J までは、複数アイテムの同時リサイズはできない。しかし、表を含むページを EPS ファイルでページ保存し、その画像をドキュメントに貼り込めば、拡大・縮小は自由自在になる。

③ XTension を活用して拡大・縮小

バージョン 3.3J でも「QX Tools」など複数アイテムの同時リサイズを可能にする XTension はあり、これらを使えば表全体の拡大・縮小も可能だ。

また全体の拡大・縮小には向かないが、ボックスの一括リサイズに大いに活躍する、AppleScript を使ったフリーの XTension 「QX BOX」などもある。

メジャーパレットを駆使して

作表にあたってぜひ駆使したいものにメジャーパレットがある。ラインやテキストボックスの位置はメジャーパレットで座標位置を正確に合わせたい。

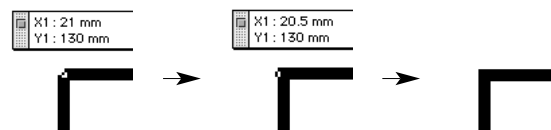
マウス作業だと、モニタ上ではきれいに見えても、プリントしたらラインが突き出していたとか切れていたということがよくある。逆にメジャーパレットでキッチリ作ってあれば、モニタ上で突き出ているように見えても、

プリントすればまず問題はない。

コピー操作が多いので、最初に位置をしっかりと合わせれば、あとはそれほど手間ではない。逆にこれを怠るとあとで余計な手間がかかることになりかねない。

もう一つ、ラインツールで作表する場合のネックに、外枠の角の処理がある。太いラインは座標位置を合わせても角が階段状になってしまう。

ラインは中心線から両側に太くなるので、縦線か横線のいずれかを、太さの半分だけ両端を伸ばすか、あるいはラインを内側に入れることで、キッチリとした角にすることができる。私はおおむね 1Q 以上の太さのラインでこの操作を行っている。



階段状になった 4Q (1mm) の太さのラインの角。横線両端を太さの半分の 0.5mm 伸ばすときれいな角になる。上のラインのサンプルはモードを「両端」にして操作している。

なお、この点では XTension 「Base&Box」の「罫線付加」も有効らしい。

※

QuarkXPress での表組み作業は、いろいろ工夫をして多少は楽になったとしても、やはり面倒な作業であることに変わりはないだろう。

しかし考え方を少し変えてみると、表とテキストボックスを別々に作るということは、どこにでも好きな位置に文字やデータを入れることができるということ。そして手作業なら XTension の機能に制約されることもない。

つまり作業は面倒で不自由だけれども、それは同時に自由な作表が可能になるということでもある。どうやら自由と不自由は紙一重なのかもしれない。

実際、私のところにやってくる仕事では定型的な表よりも、イレギュラーがあったり、むしろ記入フォームといった類も少なくない。似たようなケースや、データもそう多くなくて XTension を使うまでもないといった場合などに、紹介した方法が多少でも役に立てば幸いだ。

●えでいっとはうすのホームページ別館 QuarkXPress 奮戦記
URL: <http://ha3.seikyone.jp/home/ogn/>

第4回

キーワードは「直しやすいデータ」

スタイルシートをうまく使おう

QuarkXPress をうまく使うポイントは何だろうか。このソフトを使い始めた当初から、そして5年以上がたった今でも、それはマスターページとスタイルシートの二つではないか、と私は思っている。

この二つに共通するのは、あるパターンを作って登録すればそれを利用して繰り返し同じものを簡単に作ることができるという作成時のメリットと、元を直せばそれを利用しているデータを一括して変更することができるという訂正時のメリットにある。

印刷物はデータ作成の段階で数回の校正を経て印刷工程にまわることになるが、その過程で大小さまざまな変更が加えられる。編集者曰く――。

「章のタイトルが変わったから柱を全部入れ替えろ」

「もう少し全体のページ数を減らしたいから1ページの行数を増やせ」

「見出しのフォントやサイズを全て変更しろ」

こんなことは日常茶飯事だ。少しでもいいものを作ろうとすれば理解できないことではないが、オペレーティングする者にとっては実に迷惑なことでもある。

「えー!? この期におよんでこんなに直すんですかあ?」

と口では抵抗するが、うまく作っておけば実はそれほど大変な作業をせずに済むことが少なくない。

連載第2回で紹介したようなマスターページの作り方をしていれば、柱の変更などはまず朝飯前だ。ページの基本組みスタイルの変更も、マスターページとスタイルシートの変更で対応すれば、文字組みを維持したままで問題なく変更できる。見出しのフォントやサイズなどに至っては、スタイルシートの変更だけで簡単に一括変更

が可能だ。

で私は、「こんなこともあろうかと手を打ってたんだよね」と、密かにほくそ笑んでいるのだ。

連載1回目のテキストボックスの作り方、2回目のマスターページの作り方は、私流の直しやすいデータ作成の基礎だった。そこで今回はもう一つの基礎、スタイルシートを考えてみようと思う。

スタイルシートはバージョン4.0になって大幅に改良された。特に、段落単位でしかスタイル登録できなかった3.3までの仕様から、文字単位のスタイル登録が可能になったことは大きい。

ところが私はいまだ4.0を使っていない。この連載は「奮戦記」なので基本的に私の体験だ。だから4.0について体験的に書くことができない。本来ならこのテーマは連載3回目に取り上げるべきだったが、そんなわけで少し躊躇があった。

しかし避けては通れないテーマなので、あえて3.3を中心にここで取り上げようと思う。4.0の体感を具体的に書くことはできないが、データ作成上での考え方の問題としては同じだと理解していただければ幸いだ。

スタイルシートって何?

スタイルシート自身はめずらしい機能ではなく、ワープロソフトなどでも広く使用されている。3.3のマニュアルの用語解説には「段落に対して一度に適用できる文字属性や段落書式の設定シート」(「ユーザーガイド」用語9P)と書いてあるが、この中で「一度に適用できる」と

というのがミソだ。

右のダイアログサンプルを見れば一目瞭然のように、文字や段落には、たとえば文字の設定だけをとってみても、フォント、サイズ、カラー、シェード、変形、トラッキングやカーニング、ベースラインシフト、文字飾りと、さまざまな要素がある。もちろんこれらを一つ一つ設定することも可能だが、一括して設定すれば作業が楽なこととは言うまでもない。

スタイルシートは、テキスト編集にかかわる文字設定、段落書式、段落罫線、タブ設定の情報を登録して、それらをまとめて段落に適用することを可能にするものだ。とりわけ、同じスタイルを繰り返し使用する場合には大いに有効な機能だ。

バージョン 4.0 では、3.3 までドキュメント単位でしか設定できなかった「ぶら下がり」が段落書式に加わっているの、これもスタイルシート登録できる項目となった。また、文字スタイルシートが採用されたことでスタイルシートパレットも段落と文字の 2 段になっている。

スタイルシートの登録

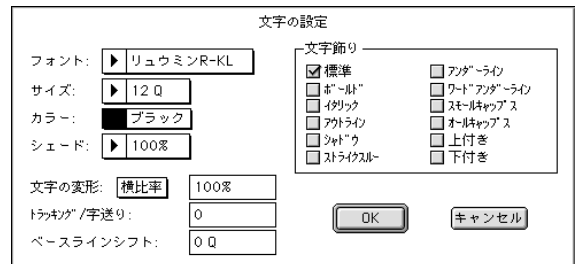
スタイルシートへの情報の登録は、編集メニューの「スタイルシート」で項目ごとに一つずつ設定するが、より簡単なのは次の方法だ。

- ①段落に一連の設定をする。
- ②その段落を選択して編集メニューの「スタイルシート」を選ぶ。
- ③開いたダイアログで「新規」をクリックする。
- ④適当な名前をつけて保存する。

これで選択している段落の情報がそのまま反映される。一度設定したものを他でも利用したいと思ったら、こうしてサッと登録すればいいわけだ。もっともこの場合、登録のために選択した段落はノーマルなど元のスタイルのままなので、新たに作ったスタイルを適用しておくことを忘れずに。

また文字の設定や段落書式には、それぞれのパレットを見ているだけではつい見落としとしてしまいそうな情報も登録できるから、けっこう奥が深い。

たとえば文字の設定の「フォント」。ここにはフォン

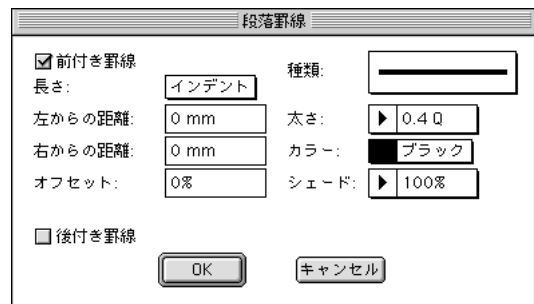


文字設定のダイアログ

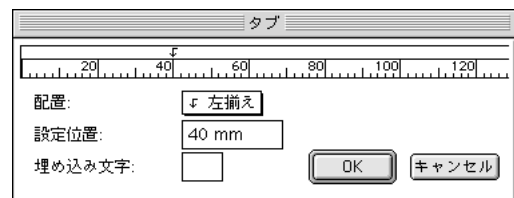


段落書式のダイアログ

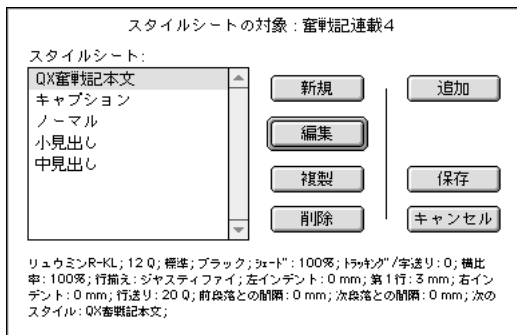
これは文字設定のダイアログとともにこのページの実際の本文スタイルで、12Qの文字を使っているの「第1行インデント」は3mm、横組みなので「ベースライングリッド固定」を使用し、H&Jは追い込み用のオリジナル設定を使用している。ひょっとしたら「横組みでなぜ文字揃えがベースラインでないのか」と思われる方があるかもしれないが、私はセンタリングでも特に問題ないと思っている。



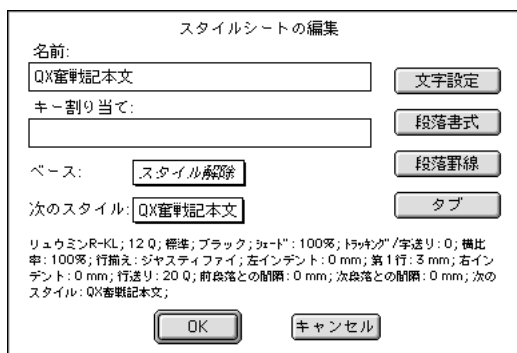
段落罫線のダイアログ



タブ設定のダイアログ



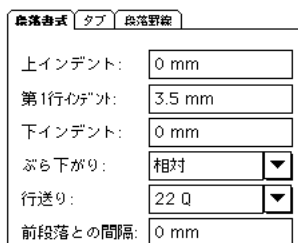
編集メニューの「スタイルシート」で開く登録ダイアログ。これはこのページで使用しているもの。



スタイルシート編集ダイアログ。文字設定、段落書式、段落罫線、タブが登録できるのがわかる。主な情報は下欄に表示されている。



バージョン 4.0 には文字スタイルシートが追加されたので、スタイルシートパレットは、段落と文字の 2 段になっている。



バージョン 4.0 の段落書式ダイアログ。「ぶら下がり」が追加されているのがわかる。

トセットの情報も登録できる。フォントセットを作って登録すると、それはフォント一覧の一番上に表示され、フォントの選択項目として選ぶことができるからだ。

同じように別のメニューで設定値を登録したものを選択できるのは、ほかに「カラー」と段落書式の「H&J」がある。特にバージョン 4.0 で禁則レベルに強弱をつけたりカスタマイズすることが可能になっているが、この選択は「H&J」のところで行うようになっている。

これらをスタイルシートに登録することで、複数の設定をスタイルによって使い分けることも可能だし、それぞれの設定値を変更すれば、それを使用しているスタイルシートも連動して変更される。

スタイルシートの適用

さて、便利なスタイルシートだが、だからといって何でもかんでもスタイル登録するのも考えものであるようだ。私は当初、とにかく複数回利用するものをみな登録していたものだが、あまりに数が多くなるとかえって混乱してしまった。

これはたとえば、ユーザ辞書にたくさん単語登録したものの、登録した簡潔な「読み」を忘れてしまうことと似ている。ちなみに私は辞書登録の場合、省略しないですそのままの「読み」を登録するようにしている。これなら忘れない。スタイルの場合はできるだけ、誰が見ても理解しやすい名前をつけておくのがいいだろう。

また見出しなども、スタイルがほぼ共通している書籍などでは大いに有効だが、さまざまな見出しを使用するチラシや新聞類、雑誌などでは、スタイルシートを使用する効果が少ない場合もある。

だから、作成するものによってスタイルシートの活用を使い分けることが肝要だが、そのためにも、スタイルシートの適用について理解しておくといいと思う。

スタイルシートの適用は、適用したい段落を選択した上で、スタイルメニューの「スタイルシート」で適用するか、スタイルシートパレットの該当スタイルをクリックする。言うまでもなく、パレットを使用するのが簡単だ。キーを割り当ててショートカットを作ることも可能だが、私はもっぱらパレットクリックで行っている。

ところが、こうしてスタイルを適用したのに、変わらないケースがある。それは次のような仕組みのためだ。

まず、デフォルトで必ず「ノーマル」というスタイルが適用される。ある段落を選んで、パレットでたとえば「本文」というスタイルをクリックすると、そのスタイルに変わる。その後別のスタイルに変更したければ、別のスタイルをクリックする。ここまでは問題ない。

スタイル適用ではなくフォントだけを変えるなど部分的な変更を施すと、パレットでクリックしても完全にそのスタイルにはならない。フォントを変更していれば、フォント以外の項目が適用されたスタイルに変わる。だから一見すると「変わっていない」ように見えるわけだ。この仕組みは、マスターページアイテムの「変更箇所保持」とほぼ同じとみて差し支えないだろう。

この時スタイルシートパレットには、スタイル名に「+」がついて表示されるので、見分けることができる。

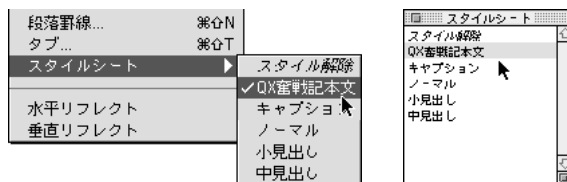
完全にスタイルを適用するには、オプションキーを押しながらスタイルシートパレットのスタイル名をクリックする。すると、適用されていたスタイルがクリアされて、新たなスタイルに置き換わる。

この仕組みを理解しておけば、ケースによって完全な適用と部分変更を活かした適用とを使い分けることが可能だ。また、スタイルシートの設定を変更しても部分変更は維持されるから、基本的なスタイルを作っておき、必要に応じて部分変更を加える使い方も有効だ。このあたりはオートカメラの絞りとシャッタースピードの関係のように、書式を優先するか文字設定を優先するかなど、ベースを決めてかかればいだろう。

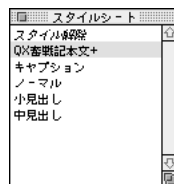
よく使うスタイル

どんなものをスタイルシートに登録するかはまったく自由だし、これといった決まりはない。基本的には複数の箇所で使用されるパターンを登録しておけば作業は楽になる。登録するかどうか迷った時に私は、後で一括して直す可能性があるかどうか、をその基準にしている。

私がよく使うスタイルは、まず本文用のスタイル。これは必ず作っている。基本的には段落冒頭の第1行インデントを伴うものだが、必要に応じて、それのないもの、



スタイルシートを適用するメニュー（左）とパレット（右）



テキストに変更を加えると、適用しているスタイル名に「+」がついて表示される。

段落	
左インデント:	3 mm
第1行インデント:	-3 mm
右インデント:	0 mm
<input type="checkbox"/> ベースライングリッド固定	
<input type="checkbox"/> ドロップキャップス	
行揃え:	ジャスティファイ
文字揃え:	センタリング
H&J:	追い込み

- ①段落に一連の設定をする。
- ②その段落を選択して編集メニューの「スタイルシート」を選ぶ。
- ③開いたダイアログで「新規」をクリックする。
- ④適当な名前をつけて保存する。

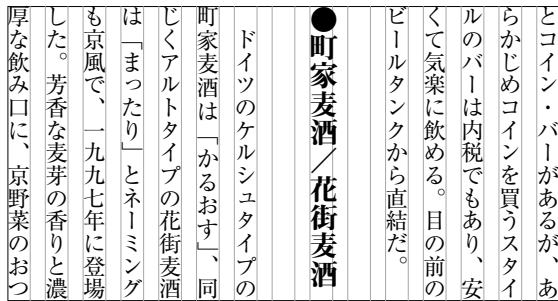
「左インデント」と「第1行インデント」を組み合わせれば上のような箇条書きも簡単。サンプルは文字サイズが12Qなので3mmと-3mmの組み合わせだが、これで文字の追加や削除に何のストレスもない。

考え方は簡単で、「左インデント」がベース位置、「第1行インデント」はそこを基準にした相対位置となる。

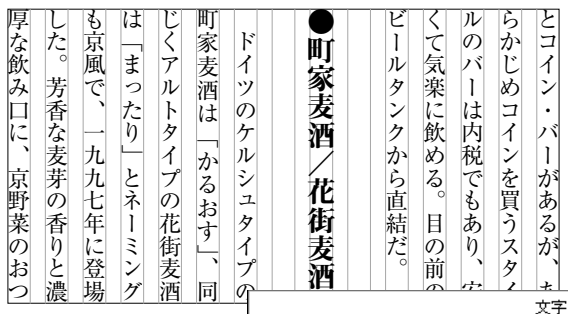
段落	
左インデント:	9 mm
第1行インデント:	-9 mm
右インデント:	0 mm
<input type="checkbox"/> ベースライングリッド固定	
<input type="checkbox"/> ドロップキャップス	
行揃え:	ジャスティファイ
文字揃え:	センタリング
H&J:	追い込み

- ①段落に一連の設定をする。
- ②→ その段落を選択して編集メニューの「スタイルシート」を選ぶ。
- ③→ 開いたダイアログで「新規」をクリックする。

「左インデント」と「第1行インデント」の組み合わせによって行頭が「左インデント」よりも左にきた場合、この「左インデント」はタブ設定と同じ働きをする。上のサンプルは左の設定で、②③はタブを入れた例。

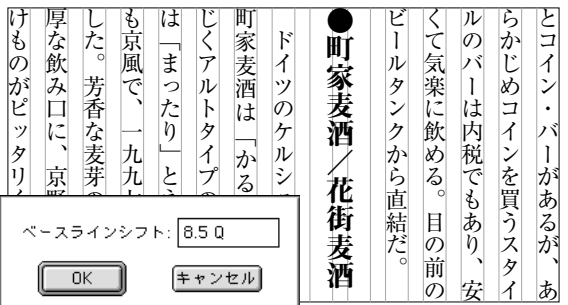


見出しの文字サイズを本文と異なるサイズにすると、センターからずれる。これは縦組みの場合は左端、横組みの場合は下端を基準に大きさが変わるため。



本文と見出しの文字サイズの差の半分をベースラインシフトですらすすることで、見出しをセンターに配置することができる。
サンプルは本文 12Q、見出し 15Q なので、その差 3Q の半分である 1.5Q をマイナスした。もちろん単位が pt でも同じこと。

フォント:	▶ 見出ミンMA31
サイズ:	▶ 15 Q
カラー:	▶ ブラック
シェード:	▶ 100%
文字の変形:	縦比率 100%
トラッキング / 字送り:	0
ベースラインシフト:	-1.5 Q



2行センターの見出しは行送り値の半分をベースラインシフトですらするが、これも見出しのサイズが違えば本文文字サイズとの差を調整する必要がある。
サンプルは行送り 20Q で文字の関係は上のサンプルと同じ。2行のうち左側にテキストをおくと、20Q の半分に先の -1.5Q を加えて 8.5Q のシフト値となる。

カギ括弧などの半角下げ用、引用などで1~2字下げをするもの、簡条書きなどで番号などの頭出しをする場合など数パターンを作る。見出し用も、大見出し、中見出し、小見出しなど数パターンを作る。その他キャプション用なども必ず作るもののひとつだ。

さて、スタイルというよりもそれを構成する文字設定や段落書式のことになるが、スタイルシートに登録するメリットの大きいものと、その設定上の留意点を少し紹介しよう。

①第1行インデント

本文用の段落書式は、ほとんどの場合に「第1行インデント」を使用している。

段落冒頭の処理については、主としてスペースを使用する方法とこの「第1行インデント」を使用する方法とで、人や会社によって考え方が異なるようだ。

私は、ベタ組みをする場合に「第1行インデント」、詰め組みをする場合はスペースというように使い分けている。ほとんどがベタ組みなので「第1行インデント」が多くなる。

ただ、スペースを使う場合は注意が必要で、とりわけフロッピーなどでデータ入稿された場合は曲者だ。これらのスペースが全て全角なら問題はないが、半角スペース2個というケースがよくあるからだ。ワープロ専用機でデータを作成しMS-DOS変換すると、全角スペースをわざわざ半角スペース2個に置き換えるものもある。そして半角スペース2個はQuarkXPress上で全角スペースと同じにはならないのだ。そのため、段落冒頭の字下げが正しくできないことがある。

こうしたトラブルを避けるにも、段落冒頭のスペースを全て削除し、「第1行インデント」を使用するのが安全だと思っている。

②引用や簡条書きのインデント

引用などは通常、本文に対して1~2字下げが施される。ところがこれまたデータ入稿された場合に、これをスペースで行っているものが少なくない。簡条書きなどの場合も同様だ。

これは段落書式の「インデント」を使用するのがよい。

スペースで行うと訂正の場合に大変な手間になるのは想像に難くないだろう。「第1行インデント」も含めてレイアウトソフトだけが持つ特殊な機能でもなく、ワープロソフトでも採用されているものなのだが、どうしてこの機能を使わないのだろうかと疑問に思うことがしばしばある。

インデント機能の使用例をサンプルで少し紹介しておこう。

③見出しの位置調整

本文に対してフォントサイズの異なる見出しを入れる場合、注意が必要だ。QuarkXPressは、フォントサイズを大きくしたり小さくすると、文字のセンターを中心に両側に太ったり細ったりするのではない。片方向、つまり横組みの場合は下を基準に上側が、縦組みの場合は左を基準に右側が大きくなったり小さくなったりする。このため見出しの両側のスペースが不揃いになって、センターからずれることになる。

そこでベースラインシフトで調整する必要がある。その調整値は、本文と見出しの文字サイズの差の半分だ。たとえば本文が12Qで見出しが15Qだとすると、見出しが3Q分上(または右)に太る。そこでその半分の1.5Q分下(または左)にシフトするわけだ。つまり、

シフト値 = (本文文字サイズ - 見出し文字サイズ) ÷ 2
となり、先の例は $(15 - 12) ÷ 2 = -1.5Q$ となる。これでバッチリ。

④2行センターの見出し

2行取りのスペースのセンターに見出しを配置する場合もベースラインシフトでの調整が必要だ。見出しと本文が同じ文字サイズなら、単純に行送り値の半分の値をシフトすればよいが、③の例のように見出しと本文のサイズが違えば、これも加味する必要がある。つまり、2行の左側行の見出しテキストを右側にずらすとして、

シフト値 = 行送り値 ÷ 2 + (本文文字サイズ - 見出し文字サイズ) ÷ 2

となり、たとえば行送りが20Qで見出しと本文の文字サイズの関係が先と同じであれば $20 ÷ 2 + (15 - 12) ÷ 2 = 8.5Q$ となる。これでバッチリだ。

このあたりの計算をやりやすくするには、使用する文字サイズや行送り値に端数のあるサイズを使わないことだ。そうすれば暗算でもすむ。もちろんダイアログに計算式を打ち込めばより正確だ。

これらはいずれも、その都度設定するのはけっこう面倒だが、一度設定してスタイルシートに登録すれば、あとは何のストレスも感じることなく適用できるし、変更があってもスタイルシート一つを変更すれば一括訂正できるから、訂正のストレスも少ない。

そのためには、最初に基本組みを作る時に、テキストボックスのサイズに文字サイズや行送りを合わせるのではなく、文字サイズや行送り値に合わせたテキストボックスにするのがよい。だから連載第1～2回で紹介したテキストボックスやマスターページの作り方は、こうしたところに役立つ方法でもあるのだ。

紙数の関係で今回は紹介できなかったが、段落書式の「前段落との間隔」「次段落との間隔」や、「H&J」を使って追い込みや追い出しを使い分けたり、面白い使い方ができる機能は他にもある。

こうしてスタイルシートは、効率的な作業と直しやすいデータを作るために大いに役立てたい。

ただスパンの長い仕事で、久しぶりに校正が返ってきたら、自分が何をどこに適用したのか忘れてしまっていた、という経験がある。そうになると、いちいちチェックして調べるのは手間。スタイルシートも登録し過ぎると、まさに「過ぎたるは及ばざるがごとし」となりかねないからほどほどに。

●このページの制作環境
PowerMac8500/G3-250MHz
MacOS8.1
QuarkXPress3.31r7
フォント=リュウミンR-KL、見出しミンMA31、新ゴL、新ゴM、新ゴB
使用した主な設定=文字単位Q、基本文字サイズ12Q、行送り20Q、ベースライングリッド固定、標準emスペースON、ぶら下がりOFF、和文/欧文のスペース50%、H&J「追い込み」の設定値：標準設定のかな間隔と漢字間隔の最小値をそれぞれ-10%に変更

●えでいっとはうすのホームページ別館 QuarkXPress 奮戦記
URL: <http://ha3.seikyounet.jp/home/ogn/>

第5回

知っている「ちょっと得する」Tips

各種設定に隠れた楽チン操作

QuarkXPress3.3J は無駄な機能がほとんどないアプリケーションソフトだと思う。それでも中には、設定ダイアログなどを見ても「これって何?」とやりすごしてしまうものがある。ところが、こうした一見よくわからないものの中に、楽チン作業の種が隠れていたりするから面白い。

もうご存知かもしれないし、これまで小出しにしたことと重なる部分もあるが、今回は、私が気づいたり教えていただいた QuarkXPress の「ちょっと得する」Tips を紹介しようと思う。もちろんマニュアルにも書いてあることなので、これらは「裏技」とは言わない。

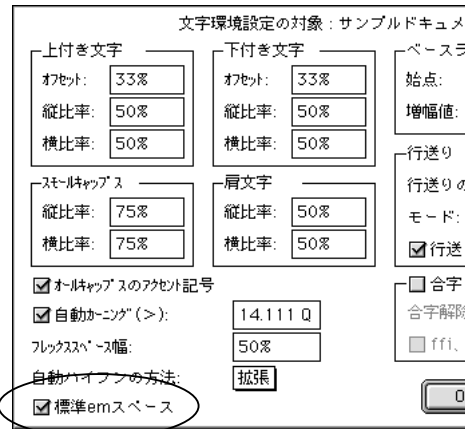
「標準emスペース」は要チェック

文字環境設定ダイアログの左下に「標準 em スペース」というチェックボタンがある。バージョン 3.3 になって追加された機能だが、私は「標準 em スペースって何?」という状態だったので、長い間、デフォルトのままチェックしないで使用していた。これは大きな不覚だった。

■同じトラッキング値なのになぜずれる?

デフォルトだと、次のような問題が発生する。

たとえば4文字と3文字を揃える均等割付をしようと思う。トラッキング・カーニングは「200分の1」を単位として調整するので、これを「200」にすると全角幅ということになる。そこで3文字中の2文字のトラッキングを100(半角空き)にして揃えようとするのだが、明朝系フォントでは問題ないものの、ゴシック系フォ



文字環境設定ダイアログの左一番下にある「標準 em スペース」のチェックボタンは、デフォルトではチェックされていない。ついそのまま使ってしまうがちだが、チェックすると効果大だ。

「標準 em スペース」のチェックがオフの状態では、同じトラッキング値でもフォントによって幅に違いが出る。そのため私は、均等割付をするのに、テキストボックスを別にして強制割付をする(むしろ、この方法が有効な場合も少なくない)など難儀していた。

「標準 em スペース」チェック OFF	
各下段「交際」にトラッキング 100	
見出しミンMA31	消耗品費 交際費
DFP平成ゴシック体W7	消耗品費 交際費
じゅん101	消耗品費 交際費
新ゴB	消耗品費 交際費

トでは広がってしまうのだ（前ページサンプル）。

私は、よくわからないまま仕方がないものとあきらめ、目測で調整するか、あるいは別にテキストボックスを作って強制割付をするという方法で作業をしていた。

■ 1em を文字サイズにする機能だった

ある時、この「標準 em スペース」のチェックボタンが気になって、試しにチェックしてみた。テキストがリフローした感じがしたので「ん？」と思って、マニュアルを開いてみた。するとそこには次のように書かれていた。

「QuarkXPress では、フォントのゼロ 2 つ (00) 分のスペースを 1em として定義していますが、標準 em スペースチェックボタンをオンにすることにより、ゼロ 2 つ分ではなくテキストのポイントサイズと同じサイズが em スペースとなり、トラッキングやカーニングの際の計算基礎として使われます」

(QuarkXPress3.3 日本語版新機能マニュアル p24)

つまり、「0」の幅はフォントによって違うから 1em の幅もフォントごとに違うことになり、それが揃わない原因だったのだ。さっそく「標準 em スペース」をチェックして均等割付を試してみると、「おお！」なんと簡単にきれいに揃うではないか（サンプル）。

結局これは、1em を文字サイズと同じにする、つまり、トラッキング・カーニング値 200 をすべてのフォントで全角に合わせる機能だった。ならば「1em を文字サイズと同じにする」などというチェックボックスにしてくれればもっとわかりやすいものを、と思う。

なお、これは環境設定項目なので、作業の途中で変更すると文字組に影響する可能性があるから要注意。

※一般に「em」は大文字の「M」の幅を基準にしたものでほぼ全角、「en」は同じく「N」の幅を基準としたものでほぼ半角、とされている。また同じ幅に対して「1em」と呼んだり「200em」と呼んだり、ソフトなどによって違いもあるようだが、ここでは QuarkXPress の定義に準じた。

分離禁止フレックススペースで禁則拡大

文字環境設定ダイアログで先の「標準 em スペース」の

見出しミンMA31	00
DFP平成ゴシック体W7	00
じゅん101	00
新ゴB	00

ゼロ二つ「00」の幅は、フォントによってこんなに違う。これを基準にすれば、フォントの違いで揃わないのは当然だと納得できる。

「標準 em スペース」のチェックをオンにすると、フォントの違いにかかわらず、トラッキング・カーニングの数値設定でねらった通りに揃ってくれる。均等割付に限らず、約物をはじめさまざまな場面の空きや詰め処理にも大いに有効だ。

「標準 em スペース」チェック ON
各下段「交際」にトラッキング 100

見出しミンMA31	消耗品費 交際費
DFP平成ゴシック体W7	消耗品費 交際費
じゅん101	消耗品費 交際費
新ゴB	消耗品費 交際費

すぐ上にある「フレックススペース幅」という項目も、私には長い間「これって何？」だった。

このフレックススペースなるものが、なんと禁則処理に役立つということを知ったのは最近のことだ。

QuarkXPress3.3J までは、禁則の対象が句読点などの約物に限られているため、促音や音引きなどは手で調整するしかない。バージョン 4.0 になって禁則対象が強化されカスタマイズも可能になったが、3.3 でも分離禁止フレックススペースを利用することでそれが可能になる。

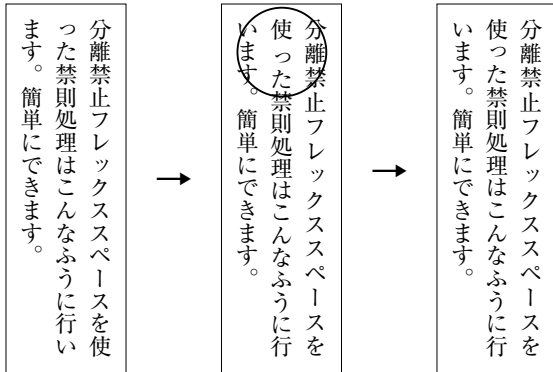
具体的な手順は右ページのサンプルの通りだが、ポイントは二つ。①促音など処理したい文字の前に「コマンド+オプション+シフト+スペース（分離禁止フレックススペース）」を入れること、②そのスペースに文字環境設定の「フレックススペース幅」で定義された数値のマイナス値でトラッキングし、幅を 0 にすることだ。

これまで、ドキュメント上で行頭禁則などを見つけたときにトラッキングで詰めるなどしていたが、校正で

<input checked="" type="checkbox"/> 自動かっこ()(>):	14.111 Q
フレックススペース幅:	50%
自動ハイフンの方法:	拡張
<input checked="" type="checkbox"/> 標準emスペース	

「フレックススペース幅」の設定は「標準 em スペース」チェックボタンの二つ上にある。デフォルトでは 50 % になっている。

●分離禁止フレックススペースを使った禁則処理の手順



- ① 2 行目の行頭に促音「っ」がきている。
- ② 「っ」の前にコマンド+オプション+シフトを押しながらスペースを入れる。これが分離禁止なので「使」が追い出され、その後にスペースが入っているのがわかる（円内）。
- ③ フレックススペースも幅をもっているので、トラッキング-50（文字環境設定の「フレックススペース幅」が 50%なので、そのマイナス値）を適用して幅 0 にする。これで OK。

●自動禁則処理のために付加すべき XPress タグ

あ → <t-50><¥!q><\$t\$> あ	ア → <t-50><¥!q><\$t\$> ア
い → <t-50><¥!q><\$t\$> い	イ → <t-50><¥!q><\$t\$> イ
う → <t-50><¥!q><\$t\$> う	ウ → <t-50><¥!q><\$t\$> ウ
え → <t-50><¥!q><\$t\$> え	エ → <t-50><¥!q><\$t\$> エ
お → <t-50><¥!q><\$t\$> お	オ → <t-50><¥!q><\$t\$> オ
っ → <t-50><¥!q><\$t\$> っ	ツ → <t-50><¥!q><\$t\$> ツ
ゃ → <t-50><¥!q><\$t\$> ゃ	ヤ → <t-50><¥!q><\$t\$> ヤ
ゅ → <t-50><¥!q><\$t\$> ゅ	ユ → <t-50><¥!q><\$t\$> ユ
よ → <t-50><¥!q><\$t\$> よ	ヨ → <t-50><¥!q><\$t\$> ヨ
ー → <t-50><¥!q><\$t\$> ー	

※ 上記はデフォルトの禁則対象以外の促音と音引きだけだが、対象文字は何でもよいので、禁則対象の自由なカスタマイズが可能だ。

●付加する XPress タグの解説

<¥!q> …フレックススペース
 <t-50> …トラッキング：-50（数値は文字環境設定の「フレックススペース幅」に合わせる）
 <\$t\$> …トラッキング解除

位置がずれると再調整が必要になることがあった。この方法を知ってからずいぶん作業が楽になった。

■融通のきくスペース

ではフレックススペースとはいったい何か。マニュアルには「ユーザーが変更できる en スペースのことです」とある（リファレンスマニュアル p69）。どうやら欧文用の機能で、ようするに「フレックスタイム」などと同じ「融通のきくスペース」という意味のようだ。

標準の欧文スペースは en の半分で、デフォルトではこのフレックススペースもそれと同じに設定してあるが、これを 1%～400%の間で 0.1%きざみで自由に設定できるという（マニュアルには「0%～400%」とあるが、実際には 0%は受け付けられない）。たとえば欧文の単語間のスペースを特定の箇所に変更する場合などに使うのだろうか。通常分離可能なフレックススペースはオプション+シフト+スペースで入力する。

もっとも、禁則処理のためにはこの幅はたいした重要性はなく、もっぱら「分離禁止」というもうひとつの属性に注目するわけだ。だから文字環境設定で定義する幅はいくらでもよいが、マイナスするトラッキング値はその定義値と同じである必要がある。

■XPress タグで自動禁則処理

あらかじめテキストに分離禁止フレックススペースの XPress タグを付加してからドキュメントに流し込めば、自動禁則処理が可能になる。

付加すべきタグは左の一覧の通り。タグの付加作業は、「ちかん君」（野本夏俊氏作のフリーウェア。ダウンロードは <http://www.linkclub.or.jp/~nomo/> より）など一度に大量の検索・置換を行えるツールを使用すると、あっという間に出来上がる。

ただし、流し込んだテキストに新たなスタイルを適用し直すと、分離禁止フレックススペースのトラッキング設定が解除されてスペース幅が出現するので要注意だ。

私は XPress タグに詳しくなく、いつもはドキュメント上でスタイルを適用しているので、小見出しなどでこのケースに遭遇した。そのため実際にはこの方法を使用してはいない。しかし効果は確認しているから、XPress

タグに精通していれば効果的な活用ができるだろう。

「H&J」で追い込み禁則ができる

禁則処理は、先の分離禁止フレックススペースを使用した場合も含めて、通常は追い出し処理がされる。その結果、追い出した行で文字が間延びすることになる。

こうした場合、私は長い間トラッキングなどで手動処理によって追い込んでいたが、これを自動化する手取り早い方法がある。「H&J」を使うのだが、とても簡単で重宝している。手順は次の通りだ。

- ①編集メニューの「H&J」を開き「新規」を選んで「追い込み」など適当な名前をつける。
- ②「ジャスティファイ設定」の「かな間隔」と「漢字間隔」の「最小」をそれぞれ「-10」にする。
- ③「OK」をクリックして保存する。
- ④「H&J」を段落に適用するには「段落書式」ダイアログの「H&J」で、先に作成した設定名を選ぶ。

■ジャスティファイの文字詰めに制御

ところで、この「H&J」も一見するとわかりにくいもののひとつだ。これは「ハイフネーション&ジャスティフィケーション」の頭文字をとったもので、マニュアルには次のように記載されている。

「H&J」は、ハイフネーション（行末で英単語の音節の接合部にハイフンをつけて区切る）とジャスティフィケーション（テキストの行頭、行末がびったり揃うように文字や単語間のスペースを調整すること）を制御する機能です。（リファレンスマニュアル p91）

話を「J」に絞ると、つまりジャスティファイの際に、ボックスの行長とテキスト量の関係から、行末の文字を追い込むか追い出すかを制御するものだ。デフォルトでは和文関係の「最小」がすべて「0%」だからテキストは詰まらず、行末にきた禁則や分離禁止キャラクタはすべて追い出されることになる。

しかし、かなと漢字の「最小」を詰めが可能のように「-10%」程度に設定すれば、追い込みができるようにな

ジャスティファイ設定

	最小	最適	最大
英単語間隔:	100%	100%	150%
和文約物:	0%	0%	15%
かな間隔:	-10%	0%	15%
漢字間隔:	-10%	0%	15%
英字間隔:	0%	0%	15%

実行領域: 0 mm

ジャスティファイ機能

サンプルの左は「H&J」標準、右は上の設定を使用した。かなと漢字を5%まで詰められる設定なので、全角文字の詰めは行長が20字以上で有効になる。

長い行だと分離禁止フレックススペースを使
った禁則処理も「H&J」で簡単に追い込み
できます。

長い行だと分離禁止フレックススペースを使
った禁則処理も「H&J」で簡単に追い込み
できます。

段落書式

左インデント:	0 mm	行送り:	20 Q
第1行インデント:	3 mm	前段落との間隔:	0 mm
右インデント:	0 mm	次段落との間隔:	0 mm

ベースライングリッド固定 次段落との分割不可
 ドロップキャップス ウィド"/ウオ/ファン制限

行揃え: ジャスティファイ

文字揃え: カスタム

H&J: 追い込み 標準

「H&J」の適用は段落書式で行う。もちろんスタイルシートに登録可能だから、スタイルでの使い分けも簡単。

る。これは5%までの詰めを可能にするものだ。

ただし、行長が短いと効果が現れない場合がある。特に促音や音引きなどは全角文字だから、20字以上の行長で有効（もちろん、40字なら2字詰まるということではない）。句読点などは10字程度で効果が現れる。

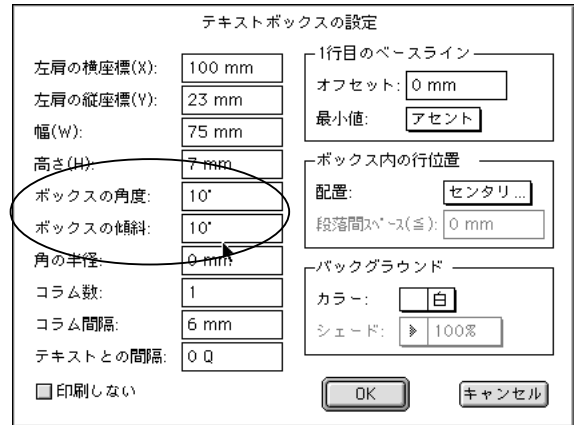
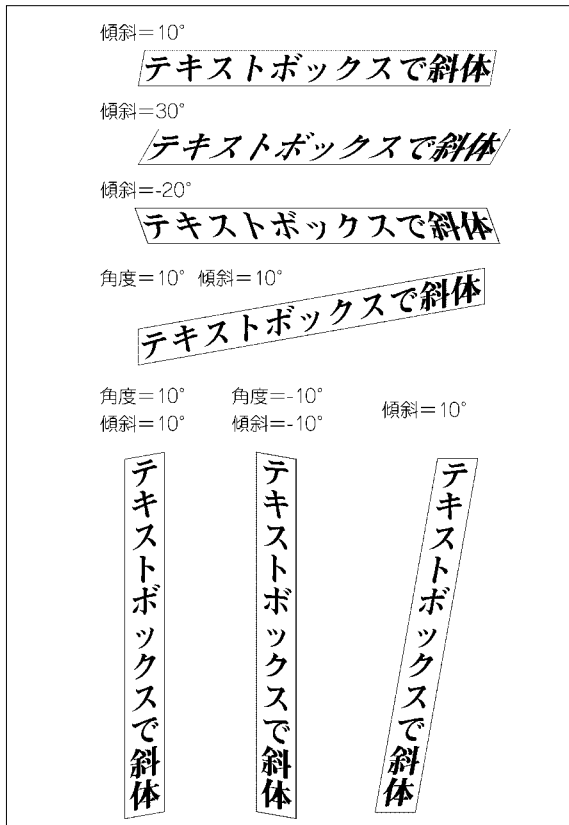
ところで、設定した「-10%」と実際に詰まる比率が違うが、私は「%」を無視し、トラッキングの「-10」と同じと考えて操作している。「和文・欧文間のスペース」も、実際には設定したパーセント値の半分になるから同様。バージョン3.3ではこの理解で差し支えはない。

しかし4.0では改善され、設定比率と実際は同じになるそうだ。これに関して私は未確認である。

■「H&J」は後ろが詰まる

追い込みの「H&J」設定で注意すべきことが二つある。

●テキストボックスの「傾斜」で文字を斜体にする



テキストボックスの角度と傾斜は「設定」ダイアログで設定する。特に「傾斜」はこのダイアログでしか設定できないので、盲点だった。

ボックスの「傾斜」で自由な斜体文字

QuarkXPress のドキュメント上で斜体など文字に角度をつけたい時、私は長い間スタイルの「イタリック」一本槍だった。それしかできないと思っていた。もちろん、ワントイプしかできないので不満だった。それ以外のものは Illustrator で作って貼り込んでいた。

ところが、テキストボックスの「傾斜」と「角度」を使って変形すると、中のテキストもそれに伴って変形する。バージョン 4.0 になってベジェ曲線にそってテキストを自由に変形することなどが可能になったが、3.3 でも直線的な変形は可能だったのだ。ちょうど Illustrator の「シアー」に近い機能だが、QuarkXPress の場合は角度が فقطで文字の大きさなどは変化しない。

左のサンプルに示したように、「傾斜」だけでもよいが、「角度」と組み合わせるとさまざまなバリエーションができるので、かなり自由に斜体文字を作ることができる。とりわけ縦組み時には重宝している。

しかし、なぜこれに気づかなかったのか。私は日頃の作業をメジャーパレットを中心に行っているが、そこから制御できる操作の中に、なぜか「傾斜」が含まれていなかったからだった。

テキストボックスを選択している場合、ボックスの「角度」の操作はできるが、「傾斜」は操作できない。画像

ひとつは「和文約物」はデフォルトのままにしておくこと。たとえばカッコと句読点が重なるなど約物が連続した場合、QuarkXpress は自動的に 50% 詰める仕様になっている。そのため、和文約物の「最小」を同じように「-10%」にすると、予想外の動きをすることがあった。だから変更しないのが安全だ。

もうひとつ、「H&J」は詰めや空きが、トラッキングと同じように文字の後ろに対して適用されること。したがって開始カッコなど、むしろその前が詰まってほしい文字に対しても後ろ側が詰まる。この仕様はいただけないのでぜひ改善してほしいところだが、残念ながらバージョン 4.0 でも同じだ。

よって、あまり大きな「最小」値にするのはふさわしくない。同様に、文字詰めに「H&J」を使用するのも考えものだ。私はこの追い込み設定くらいに限定している。

ボックスの場合は、ボックス内の画像の「傾斜」と「角度」、ボックスの「角度」が操作できるが、やはりボックスの「傾斜」は操作できない。

すべてを小さなパレットに詰め込むのは難しいだろうが、このあたりはボックスの基本的な制御に属することだからなんとかしてほしいと思う。しかし、バージョン4.0でも変わっていないようだ。

なお、「傾斜」を適用したものと同じ形は多角形ボックスでも作れるが、こちらは中のテキストが変形するわけではない。念のため。

アンカーボックスの回り込みは「0」に

最後は、ごく最近知った私にとっての最新情報。

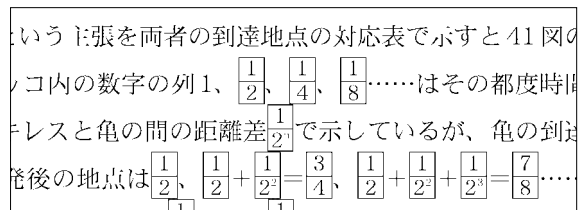
アンカーボックスは、画像ボックスやテキストボックスをテキスト中にペーストすることで、それをテキストと同様に扱うものだ。私は、右上に示したような数式を扱う組版の仕事で、このアンカーボックスと格闘することになった。

アンカーボックスの操作は、ボックスをアイテムツールを使ってカット（コピー）した後、編集ツールを使用してテキスト中にペーストする。そのままでは位置がずれることがあるので、ベースラインシフトなどでの調整が必要だが、テキストの移動に伴って動いてくれるので便利なアイテムではある。

このアンカーボックスの作業で最も困ったのが、サンプルのように複数行に連続する場合だった。アンカーボックスは複数行で重なることができない。分数が多かったので私は、行送り 25Q (6.25mm) に対しアンカーボックスのサイズを 6mm にしていた。複数行に連続しても収まるはずだ。

しかし実際には、うまくいく場合と、収まらずに 1 行空いてしまうケースとがあった。調べてみると、どうやらアンカーする前の通常のボックスの状態での「回り込み」の設定の違いに原因があることがわかった。

しかしそれは、「回り込み」を「なし」にしていた場合は収まらず、「0」にしていた場合が収まっていたのだ。話が逆ではないかと思ったが、とにかく実際の動きに即して作業をしていた。



四角で示したボックスはアンカーテキストボックス。これが複数行に連続した時、ボックスのサイズによっては 1 行飛んでしまうことがある。「回り込み」を「0」にしてからアンカーするのがよい。ちなみに、分数の線は段落罫線を使用している。これも教えていただいたものだが、分数を扱う便利な Tips のひとつだ。

仕事が終わって、Web ページに数式との格闘記をアップし、その中でこの疑問を投げかけたところ、すぐに情報をいただいた。それは、こういうものだった。

- ①アンカーボックスに「回り込み」が「なし」はあり得ない。
- ②「なし」になっているボックスは、アンカーの際に自動的にデフォルトの「1pt」が適用される。
- ③だから「回り込み」は「0」が最も扱いやすい。

これは納得できる。私のケースは、「なし」にしていたためにデフォルトに戻り、回り込みを含めたアンカーボックスのサイズが行送り値をオーバーしたため収まらなくなった、ということだったのだ。

というわけで、アンカーボックスの「回り込み」は「0」にすべしだ。これについてはマニュアルで探してみたものの、該当する記述にはまだ出合っていない。



さて、多少は役に立つものはあっただろうか。次回は最終回、トラブルシューティングを。

●このページの制作環境
PowerMac8500/G3-250MHz、MacOS8.6
QuarkXPress3.31r7、Illustrator8.0.1、Photoshop5.0
フォント=リュウミンR-KL、見出しミンMA31、新ゴL、新ゴM、新ゴB、Lじゅん101
使用した主な設定=文字単位Q、基本文字サイズ12Q、行送り20Q、ベースライングリッド固定、標準emスペースON、ぶら下がりOFF、和文/欧文のスペース50%、H&J「追い込み」の設定値：標準設定のかな間隔と漢字間隔の最小値をそれぞれ-10%に変更
フォントを含むサンプル画像= QuarkXPress のドキュメントをEPS ファイルでページ保存後、Illustrator8.0.1 でラスターライズしてビットマップ化し、貼り込んだ。

●えでいっとはうすのホームページ別館 QuarkXPress 奮戦記
URL: <http://ha3.seikyounet.jp/home/ogn/>

第6回

備えあればトラブルもまた楽し？

トラブルシューティング

自分が遭えば大汗ものだが、知人が同じ目に遭うとつて大笑いしたくなるのがトラブル。一口にトラブルと言ってもさまざまで、そのケースも環境によってまさに千差万別だ。しかし、各種の雑誌やインターネットで情報が公開・交流されているのは心強い。

Quark社のホームページ (<http://www.quark.co.jp/>) のユーザーサポート→テクニカルサポート→テクノロジーデータベースに、トラブルシューティングが掲載されている。最近のOSとの互換情報もあるので、トラブルに遭遇したらまずはチェックしてみるのがいいだろう。

また、私のホームページ (URLは記事末尾に記載) に開設している掲示板「DTPフォーラム」も含め、さまざまなDTPサイトに設置された掲示板などでも、常にトラブル情報が交流されている。掲示板の過去ログを掲載しているところも多く、それらにもあたれば、きっと類似のトラブルやその解決策に出会うことができるだろう。

ここでトラブルのすべてに触れることはとうてい不可能だが、連載の最後に、私が遭遇したり「DTPフォーラム」に書き込まれたケースなどをネタに、快適作業への道をさぐってみたい。

最大の防御はバックアップにつきる

まず基本的に、現状のDTPのワークフローの中では、程度の差はあれ、何らかのトラブルは避けられない、と考えるのが無難だろう。

もともと基本ソフトのOSや各種ソフト、フォントなどが別々のメーカーから出ている寄せ集めの作業環境な



Quark社ホームページのテクノロジーデータベース

のだから、相互に互換がはかられているといっても、何らかのトラブルが起きない保証はない。まして、最新のQuarkXPress 4.0Jでさえ1年半以上前のリリースだし、現状でまだ主流をなしている3.3Jに至っては1994年のリリースだ。それから現在までにOSをはじめさまざまな関連ソフトが何度もバージョンアップしているのだから、もはやトラブルが起こって当たり前とも言えよう。

こうなると、トラブルに遭遇してからあわてても仕方がない。トラブルによる被害を最小限に止める手だてを日頃から構築しておくことが大切だ。そうすれば、多少のトラブルなら落ちついて対処できるだろう。出合ったことのないトラブルに遭遇すれば絶好の掲示板ネタにもなったりするので、考えようによっては「トラブルもまた楽し」である。もちろん納期直前などせっぱ詰まった

時期のトラブルは願い下げだが……。

さて、トラブルと上手につきあう上でまず最小限、次の手は打っておきたい。

①十分なメモリ割り当て

Macintosh の場合、アプリケーションのメモリ割り当てを十分なサイズにすることが安定動作につながることはよく知られている。私は QuarkXPress に約 25MB を割り当てている。QuarkXPress は、Photoshop のようにメモリサイズを大きくするほど安定し動作も速くなるというソフトではないので、この程度で十分だろうと思う。

QuarkXPress を使い始めた頃はこういうことを知らず、フリーズして 1 時間の作業をパーにした、ということがよくあった。しかし最近では、メモリ割り当てが原因のトラブルケースをほとんど聞かなくなった。

②自動保存と自動バックアップ

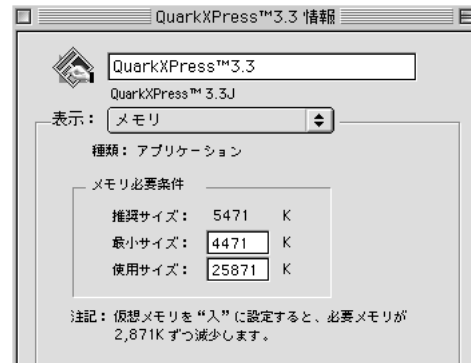
メモリを十分に割り当てていても、フリーズなどのトラブルが 100% なくなるわけではない。そこで、トラブルに遭遇した作業ファイルを救出するには、つまるところバックアップが最も確実なようだ。

バージョン 3.3J 以降は「自動保存」と「自動バックアップ」ができるようになってきているが、これは必ずチェックしておきたい項目だ。私は 10 分ごとの自動保存と 2 リビジョンの自動バックアップにしている。

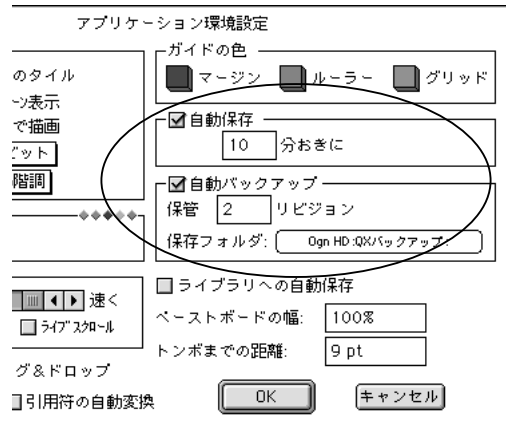
これで、フリーズしても最大 10 分間の作業をやり直せば済む。システムエラーなどを伴ってファイルが壊れた場合でも、バックアップファイルで作業を再開できるケースが多い。実はこのドキュメントも作成中に一度、なぜかファイルを失ったが、バックアップファイルで再開できたのだった。

③こまめな手動の保存

もちろん自動バックアップだけに頼らず、手動でこまめに保存するクセもつけておきたい。特に、使用ソフトを切り替える時は必ずセーブしたい。別のソフトのトラブルによるシステムエラーのために、作業中のドキュメントまで壊したくはないからだ。他にも作業の区切りや手のこんだ作業中、あるいはマシンの前を離れる時など、



メモリ割り当ては、アプリケーションアイコンを選択して、ファイルメニューの「情報を見る」で「使用サイズ」を設定する。



自動保存、自動バックアップの設定はアプリケーション環境設定で行う。

意識してセーブするようにしている。

④別ディスクへのバックアップ

さらに、ハードディスクでさえいつ何時壊れるかもしれないという可能性をはらんでいる。そこで区切りのついたファイル一式は、必ず別のディスクにバックアップするようにしている。

もちろんそのバックアップディスクでさえ完全ではないわけだから考え出せばキリがないが、少なくともこの程度の手を打っておけば、多くのトラブルに笑顔で対処できるのではなからうか。

最近よく聞くトラブル

最近、特に MacOS 8.5 以降、コンフリクトが原因と思われるフリーズやシステムエラーなどのトラブルをよく

耳にする。これらは前述の Quark 社のホームページに記載されているものもあるが、いくつかあげてみよう。

■スクロールバーでフリーズ

QuarkXPress ドキュメントで作業中、スクロールバーをつかんでスクロールしようとするときフリーズするという。これは QuickTime 4.0 とのコンフリクトが原因で、回避するには QuickTime 4.0 を使用停止にする。

MacOs 8.6 までは QuickTime 4.0 をインストールしなければ問題ないが、MacOS 9 からはこれが標準装備なので、注意が必要だ。

■「カラー」→「トラップ編集」でフリーズ

MacOS 8.5 以降、QuarkXPress の「カラー」で「トラップ編集」をクリックすると、フリーズまたはアプリケーションが終了する。QuarkXPress 4.05J でこの問題は解消しているらしいが、3.3J では起こる。しかし 3.3J でも、機能拡張ファイル「Edit Traps Fix 1.0」をインストールすることで回避できる。

MacOS 8.6 を使用している私もこのケースに遭遇したが、「DTP フォーラム」で情報を教えていただき、さっそく同機能拡張ファイルをインストールした。以下の URL からダウンロードできる。

http://www.praxisoft.com/techsupp/product_updates.html

■「プリンタフォントの有無」でフリーズ

MacOS 8.5 以降で LaserWriter 8 (J1-8.6 以上) を使用している場合、特別メニューの「プリンタフォントの有無」をクリックすると、時計アイコンが表示されたままフリーズする。私も確認した。

回避策は別のプリンタドライバを使用することだ。またこれも、バージョン 4.05 で解消しているという。

「プリンタフォントの有無」でのトラブルはこれ以外に、PS プリンタに接続せず、ソフト RIP を使用している場合、プリンタドライバのバージョンが上記以前でも、同様に時計アイコンが表示されたままになるというトラブルが「DTP フォーラム」に書き込まれた。

類似のトラブルもあり、どうやら PS プリンタと接続

しているかどうかに関係があるようにも思われるが、この件に関して私には、明確な原因や回避策が不明だ。

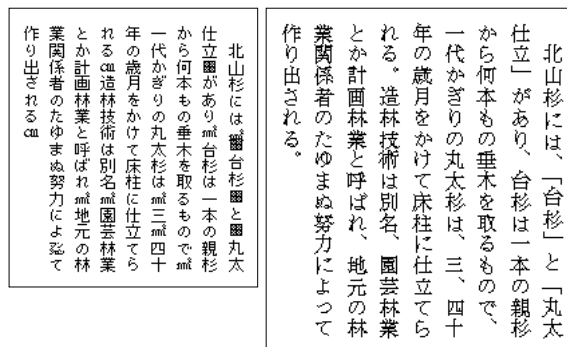
■OCFビットマップフォントの縦組み約物の文字化け

QuarkXPress に限らないが、MacOS 8.5、8.6 で、OCF ビットマップフォントの縦組み約物の表示が文字化けをする。ただしプリントには影響しないし、ATM フォントを使用すれば表示にも問題はない。だからビットマップフォントでだけの問題だが、これは MacOS 8.5 以降でのフォントの扱いの変更（丸漢ファイルをサポートしない）によるもの。

MacOS 8.1 以前のシステムでは、文字種がない文字の表示は Osaka フォントから借りていたが、MacOS 8.5 以降ではその役割を「丸漢コンパチビリティ」という特別な丸漢ファイルがするのだという。そのファイルに縦組み約物の文字種がないことが文字化けの直接の原因だ。

解決策は二つある。

- ①同丸漢ファイルに縦組み約物を追加する「丸漢コンパチ縦書サポート」をフォントフォルダに入れる。これはエヌフォー・メディア研究所からダウンロードできる (<http://www.enfour.co.jp/media/>、同社製品ユーザー以外は 3,000 円)。
- ②この問題を解消している MacOS 9 にバージョンアップする。ちなみにユーザーの実験によれば、MacOS 9 の「丸漢コンパチビリティ」を MacOS 8.5 のシステムに入れても解決したようだ。



「丸漢コンパチ縦書サポート」未使用(左)では OCF ビットマップフォントの縦組み約物が画面表示で文字化け。同ファイルを使用すると解決した(右)。このトラブルの詳しい顛末は私のホームページにも掲載している(「MacOS 8.6 とフォントの問題」参照)。

特定フォントの縦組みトラブル

フォントをめぐる QuarkXPress (3.3J 以前) 特有のトラブルに、縦組みでのリフロー問題がある。それが MacOS システム標準フォントの細明朝体、中ゴシック体と、モリサワ OCF フォントの L リュウミン L-KL、M 中ゴシック BBB との間でのトラブルだ。

サンプル左のように、細明朝体で組んだ縦組みのテキストは、L リュウミン L-KL に置き換わると文字組みが詰まってしまう。中ゴシックの場合は、M 中ゴシック BBB に置き換わると逆に広がってしまう。私も一度、リフローしたまま印刷・製本されてしまったことがある。

■リフローの3つの原因

調べてみると、この原因は次の三つだ。

- ①細明朝体と L リュウミン L-KL、中ゴシック体と M 中ゴシック BBB は、違うフォントであるのに、PS フォント名が同じであるために、システムや出力機が持つフォントの違いによって置き換わる。
- ② QuarkXPress は縦組みの文字送りに「WidMax」と呼ばれるフォントの文字幅情報を参照している。
- ③細明朝体、中ゴシック体、そしてモリサワ OCF フォントの WidMax 値のほとんどが「4096」であるのに対し、次の4フォントだけが異なる。

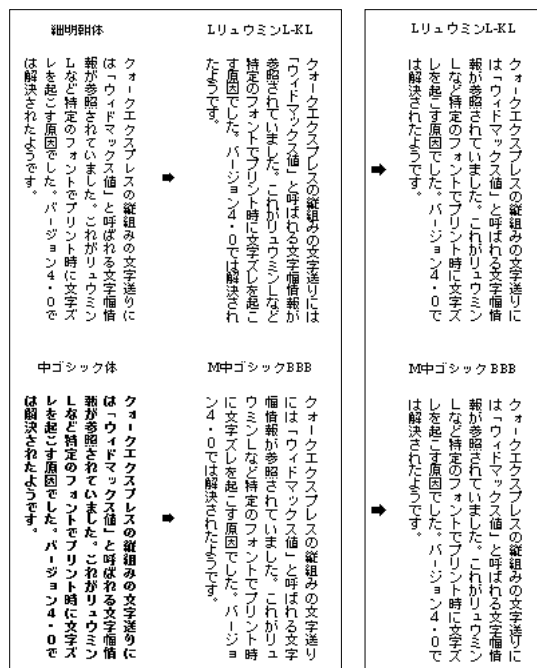
- ・ L リュウミン L-KL = 3866 (-11)
- ・ B 太ミン A101 = 3883 (-10)
- ・ 見出しゴ MB31 = 3993 (-5)
- ・ M 中ゴシック BBB = 4157 (+3)

上記のカッコ内は、WidMax 値 = 4096 を全角とした場合の、QuarkXPress のトラッキング値への換算だ。こうすれば、細明朝体から L リュウミン L-KL に置き換われば詰まるし、中ゴシック体の場合だと広がるのが納得できるのではなからうか。

■リフローの回避策

これにはいくつかの回避策がある。

- ①細明朝体、中ゴシック体、及び L リュウミン L-KL、M 中ゴシック BBB のいずれのフォントも使わない。



QuarkXPress での縦組みの画面表示。バージョン 3.3J (左) ではフォントの置き換えでリフローする。しかし 4.0J (右) では解消されている。

- ②出力機に合わせて最初から L リュウミン L-KL と M 中ゴシック BBB のみを使用する。この場合、そのままではベタ組みに工夫があるが、リフローのトラブルは避けられる。
- ③細明朝体、中ゴシック体に代えて、全体を Biblos 細明朝外字、Biblos 中ゴシック外字で指定する。
- ④ QuarkXPress 4.0J 以降を使用する。
- ⑤ CID (NewCID) フォントを使用する。

私は、出力側と確認した上で、もっぱら③の方法を使っている。ビブロス外字は、外字以外に指定してもいわゆる親文字 (L リュウミン L-KL など) に置き換わって出力され、しかもその際の WidMax 値はビブロス外字のもつ 4096 が適用されるので、リフローしないわけだ。

また QuarkXPress 4.0J 以降では、縦組みの仕様が変更され、3.3J までのようなリフローは解消されている。もっとも、3.3J で作成したドキュメントを 4.0J で開くと、基本的に 3.3J の文字組みが維持される。これを 4.0J 形式に変換するには、オプションキーを押しながらファイル

メニューの「開く」で開くとよい。

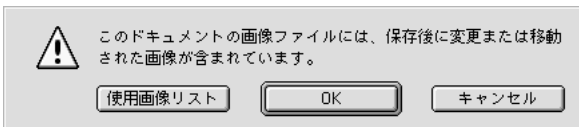
またモリサワフォントも CID 以降、M 中ゴシック BBB の 4157 を除いて、WidMax 値は全て 4096 に統一されている。

データ移動時のトラブル

フォントのリフロー問題もデータ移動時のトラブルだが、それ以外に画像ファイルをめぐるトラブルがある。

■画像ファイルのリンク切れ

画像ファイルも含めデータ一式を移動して別のマシンでドキュメントを開いた際、下のような警告が出ることもある。これは画像のリンク切れを示すもので、特に画像ファイルをフォルダに入れていると必ず起こる。確かに管理はしやすいが、データ移動時にはドキュメントと同じ階層に置くのが無難だ。



ドキュメントに画像を取り込むと、QuarkXPress はその画像の位置を、ディスクの第一階層からフルパスで記憶する。いわば住所を国名からすべて書くようなものだ。そしてそのファイルを検索する際、まず元の場所を探し、そこに該当ファイルがなければドキュメントのあるフォルダを探すのが QuarkXPress の仕様だ。

データを別マシンに移動すると、家ごと別の国に引越したようなものだから検索の出発点が変わり、元の場所がわからない。そこでドキュメントのあるフォルダを検索することになる。

だから、データの移動時には画像ファイルをドキュメントと同じフォルダの同じ階層に置くのがよいわけだ。

しかし、もしフォルダに入れていて上記の警告が出てあわてることはない。画像ファイルをドキュメントと同じ階層に出してからドキュメントを開けば OK だ。

■変更していない画像ファイルの「変更」

最近「DTP フォーラム」に書き込まれたトラブルに、

データを移動しただけなのに、画像ファイルが「変更」になったというケースがあった。

原因はマシンの日付・時刻の設定。実験して確かめた方の書き込みによると、正しい日付・時刻設定のマシンで作成したファイルを、たとえば 1 年前の設定になっているマシンに移動すると、データの修正日付がそのマシンの日付に変更されるという。この現象は時刻設定が 1 時間の違いでも確認できたようだ。

QuarkXPress がファイルを検索する際に一致情報として参照するのは、修正日付とファイルタイプ。画像ファイルの修正日付が変わったために、作成されたマシンでドキュメントに記録された情報と異なることになり、「変更」とされたわけだ。

どうやら、たかが日付・時刻とあなどらず、正しく設定しておくのがよいようだ。

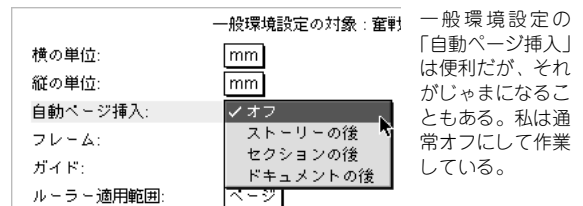
編集作業上のトラブル

最後に、編集作業中によく遭遇するトラブルをいくつか挙示しよう。

■ページが勝手に増えた

意図しないのに勝手にページが追加されることがある。一般環境設定の「自動ページ挿入」がオンで、自動テキストボックスの設定されたマスターページを使用している場合に起こる。しかも、自動テキストボックスだけでなく、ドキュメント上で追加したテキストボックスであっても、テキストがあふれるとページが追加される。

そこで私は「自動ページ挿入」を、テキストを流し込む場合だけオンにし、通常はオフにしている。



■ページが削除できない

削除したはずなのにページが削除できないケースがあ

る。該当ページにリンクされたテキストボックスがあり、そこにテキスト（リターンなどのキャラクタも含む）が残っていると削除できない。これらを確実に削除すれば、ページ削除が可能だ。

■テキストが取り込めない

テキストが取り込めないことがある。主としてワープロソフトの形式そのままのファイルだ。QuarkXPressにはワープロソフトのファイルを取り込むためのXTensionも用意されているが、それらに該当しなければ取り込めない。こうしたことから、原稿テキストは必ずテキストファイルにするのがいいだろう。



ワープロソフトのファイルを取り込むためのフィルタ。ワープロソフトのアプリケーションがなくてもこれがあれば取り込めるから、私は時々、ファイルを開くための目的で使用することがある。

また DOS テキストの場合は、改行コードの「LF」など異なる特殊キャラクタを確実に削除しておくのがよい。そのまま取り込むと、プリント時にトラブルの原因になることがある。

■文字揃えが無視される

文字揃えにセンタリングなどを適用しても無視されることがある。これはテキストにタブが含まれてケースで、文字揃えの設定にかかわらず左（上）揃えになる。この場合はタブでセンタリングすれば OK だ。

またインデントが設定されていると当然だがボックスのセンターにはこないし、ぶら下がりがオンだと行末に半字分のスペースが確保されるので、同じくセンターにはこない。ぶら下がりの場合、センタリングするには行頭にも半字分のインデントを設定する必要がある。

■ボックス内の行位置の設定が無視される

テキストボックスの上に回り込みの設定されたボックスが重なると、そのテキストボックス内の行位置の設定

は無視され、デフォルトの左（上）揃えになる。ボックス内の行位置を維持するためには、重ねるボックスの回り込みを「なし」にすれば OK。

しかしたとえば、基本組みのテキストボックス内の行位置をジャスティファイにしていると、画像を貼るためにボックスを重ねて回り込みにした際、行ズレを起こすことがある。私がこの連載で再三強調するピッチリサイズのテキストボックスでの基本組みは、こうしたトラブルを未然に避けるためでもある。

また、横組みで段落書式にベースライングリッド固定を適用しているとボックス内の行位置の設定は無視されるし、多角形テキストボックスでも同様に無視される。

■一部のテキストが間延びする

テキストの一部が間延びすることがある。テキストの行揃え設定がジャスティファイで、テキスト中に含まれるスペースが行末にくるケースで起こる。スペースが連続すれば、極端に間延びするわけだ。データ入稿で字下げなどが含まれているとよく遭遇するが、不要なスペースは流し込む前に確実に削除しておくべきだろう。

またこれは逆に、行末でスペースが活かせないということでもある。私は、スペースの代わりに適当な記号を入力してカラーをバックグラウンドと同じにするという手で逃げている。

* * * *

書き出せばまだまだあるが紙面が尽きた。長い間のつたない連載におつきあいをいただき感謝。Web ページではあいかわらず書き続けているので、興味のある方はぜひどうぞ。（了）

●このページの制作環境
PowerMac8500/G3-250MHz、MacOS8.6
QuarkXPress3.3.1r7、Illustrator8.0.1、Photoshop5.5
フォント＝リュウミン R-KL、見出しミン MA31、新ゴ L、新ゴ M、新ゴ B
使用した主な設定＝文字単位 Q、基本文字サイズ 12Q、行送り 20Q、ベースライングリッド固定、標準 em スペース ON、ぶら下がり OFF、和文/欧文のスペース 50%、H&J「追い込み」の設定値：標準設定のかな間隔と漢字間隔の最小値をそれぞれ－10%に変更

●えでいっとはうすのホームページ別館 QuarkXPress 奮戦記
URL: <http://ha3.seikyounet.jp/home/ogn/>

索引

A-X

DTP フォーラム : 33
EPS ファイルでページ保存 : 20
H&J : 7, 8, 23, 30
WidMax : 36
XPress タグ : 29
XTension : 3, 20
 Base&BOX : 3
 Guide Line Pro : 13
 QX BOX : 20
 QX Tools : 20
 テーブルワークスプラス : 17

あ

アンカーボックス : 32
1 行目のルビ : 13
インデント : 17, 18, 25, 38
 第 1 行インデント : 6, 24, 25
オーバーフロー : 9, 10

か

カーニング : 28
改行コード : 38
ガイドライン : 3
 コラムガイド : 12
 マージンガイド : 12
 マスターガイド : 12
 ルーラーガイド : 13
画像ファイル : 37
 変更 : 37
 リンク切れ : 37
環境設定 : 8
 ツール環境設定 : 17
 日本語環境設定 : 17
 変更箇所消去 : 10

変更箇所保持 : 10, 24

級数 : 5
 Q : 8, 17
行送り : 4, 16
 自動行送り : 4
 絶対行送り : 4
 増減幅指定行送り : 4
行ズレ : 18, 38
行揃え : 3, 12
禁則 : 7, 23, 28
 追い込み禁則 : 30
 自動禁則 : 29

さ

自動バックアップ : 34
自動ページ挿入 : 37
自動保存 : 34
ジャスティファイ : 4, 6, 30, 38
修正日付 : 37
スクロール : 35
スタイルシート : 21
 スタイルシートの登録 : 22
 スタイルシートの適用 : 23

た

第 1 行目のベースライン : 13, 14
縦組み : 3, 36
タブ : 19, 22, 38
段落書式 : 3, 22, 25, 30
ちかん君 : 29
テキストボックス : 4, 6, 9, 12, 16
 角度 : 31
 空のテキストボックス : 11
 自動テキストボックス : 10, 37
 傾斜 : 31
 テキストとの間隔 : 5, 16

ボックス内の行位置 : 6, 16, 18, 38
ボックスの高さ (長さ) : 7
ボックスの幅 : 4
連結 : 10, 18
枠空け : 4

特殊キャラクタ : 38

トラッキング : 7, 22, 27, 31

トラップ編集 : 35

トラブルシューティング : 33

は

バックアップ : 33

表組み : 15

標準 em スペース : 27

ファイルタイプ : 37

フィルタ : 38

フォント : 6, 22, 24, 26, 28, 35, 36

CID : 36

OCF : 35

フォントセット : 23

プリンタフォントの有無 : 35

複数アイテムの配置 : 16, 17, 18

ぶら下がり : 13, 38

フリーズ : 34, 35

フレックススペース : 28

フレーム : 16

プロポーショナル : 7

ベースライングリッド : 3, 13, 38

ベースラインシフト : 26, 32

ベタ組み : 6, 25

ポイント : 5, 7, 8

ま

マージン : 13

マスターページ : 6, 9, 10, 21

再適用 : 11

見開きマスター : 11

丸漢コンパチビリティ : 35

回り込み : 5, 9, 16, 32, 38

見出し : 26

メジャーパレット : 8, 20, 31

メモリ割り当て : 34

文字揃え : 6, 12, 17, 38

や

約物 : 7, 31, 35

横組み : 3, 13

ら

ライブラリ : 12, 13

ライン : 15

ラインツール : 16, 20

リフロー : 28, 36

小国文男 (おぐに ふみお)

1959年京都府生まれ。フリーランスライター&編集者&DTPオペレーター。経理関係の仕事を経て1993年よりフリー。著書に『ワープロしんぶんDTP編集入門』日本機関紙出版センター1993年。

ホームページ

- えでいっとはうすのホームページ

<http://village.infoweb.ne.jp/~fwbc3841/>

- えでいっとはうす別館 QuarkXPress 奮戦記

<http://ha3.seikyou.ne.jp/home/ogn/>

- DTP原稿データ入稿Q&A (共同制作サイト)

<http://ha3.seikyou.ne.jp/home/ogn/project/>

E-mail : fwbc3841@mb.infoweb.ne.jp

初出/CIL・Gordian Kont 編集委員会発行「Gordian Knot」誌
第19号(1999年1月)～第24号(2000年6月)。

PDF版 QuarkXPress 奮戦記

2000年7月1日初版発行

2000年12月1日第2版発行

2001年6月1日第3版発行

著者 小国文男

発行者 小国文男

発行所 えでいっとはうすOGN

〒616-8177 京都市右京区太秦馬塚町14-8

TEL&FAX 075-882-5407

URL : <http://village.infoweb.ne.jp/~fwbc3841/>

E-mail : fwbc3841@mb.infoweb.ne.jp

© Fumio OGUNI 2000

本書の内容の一部または全部を、個人で使用する目的以外に無断で複写・複製・転載、あるいはデータを改変することは、著作権法で禁じられています。また本PDFファイルを、不特定多数がアクセスしダウンロードできる環境に置くなどして再配布することをご遠慮ください。
本書の内容に関するお問い合わせはEメールをお願いします。

第1回●1万分の1ミリの明暗

ピッチャリサイズテキストボックスの行揃えと文字揃え効果

第2回●マスターページとプラトニック

マスターページアイテムには指を触れるな

第3回●自由と不自由は紙一重？

マニュアルな表組み作業を考える

第4回●キーワードは「直しやすいデータ」

スタイルシートをうまく使おう

第5回●知っていると「ちょっと得する」Tips

各種設定に隠れた楽チン操作

第6回●備えあればトラブルもまた楽し？

トラブルシューティング