

プリセット指定マニュアル

かんたんですが、プリセットについて解説します。今後マニュアルにまとめる計画ですので、読みづらさはしばしご容赦ください。※ ときどき、実際のプリセットファイルの内容に追いついていない場合があります。申し訳ありませんがご容赦ください（もしできればご指摘ください）。

プリセットとは / つかいかた

プリセットファイルとは、リーダとしてのデフォルト挙動やレイアウトの初期設定を外部化したテキストファイルです。これを編集することでビビの振る舞いをカスタマイズすることができます。

bibi/presets フォルダに入れて使うのが基本ですが、じつはどこか別の場所でも（余所のサーバでも）かまいません。

配布状態の bibi/index.html は bibi/presets/default.js を呼び出しています。設定を変更したいときは、この default.js を直接編集してもいいのですが、default.js を複製・リネームして別のプリセットを作成した上で、bibi/index.html から呼び出すファイルをそちらに変更することをお勧めします。

bibi/index.html も、たとえば bibi/the-great-book.html のように複製・リネームして使い分けることができるので、本ごとに HTML とプリセットの組み合わせを使い分けることができます。

カスタマイズについて、はじめに

以下、挙動をカスタマイズできる具体的な指定について説明しますが、多くは、対象とする EPUB によって一長一短があります。だからこそ、ビビ側で一律に固定せず、プリセットとして編集可能にしてあるというわけです。

読者自身が読みながらボタンで変更できる項目もありますが、ご自身で作成した EPUB を読者に見せたい場合や、ご自身で見た目をカスタマイズした Bibi を読者に使ってもらいたい場合など、EPUB に合わせたスタイルやあなたらしいスタイルを予めプリセットで指定しておく、きっとステキだと思います。

この文書内で使用される用語の定義

アイテムとは、表示されるファイルひとつひとつのことです。

「見開き」とは、アイテムをつつむブロックです。リフロー指定のアイテムはそれ 1 つだけで「見開き」を構成します。固定レイアウトで左右ペアが作られている場合は、その 2 つが 1 つの「見開き」に入ります。

「長さ」は、進行方向サイズのことです。水平方向に進行しているときは左右幅、垂直なら上下高を指します。同様に「広さ」は、水平進行なら上下高、垂直なら左右幅です。そして、それらの数値が小さい状態を「短い」「狭い」ということにします。


これらの定義は EPUB もしくは電子書籍について一般に使われているわけではなく、あくまでビビ用語だと思ってください。

項目の説明 1：基本設定

ビビのごく基本的な振る舞いを制御します。

"bookshelf" bookshelf ディレクトリまでのパス

そのプリセットファイルからみた bookshelf ディレクトリのパスです。基本的には、同じウェブサーバ内の bookshelf ディレクトリまでの相対パス・ルート相対パス（"./../bibi-bookshelf/" や "/bibi-bookshelf/" など）を記述します。

高度な設定として、クロスオリジンリクエストが許可されているサーバの bookshelf ディレクトリを http(s) / から書くこともできます。ただし、そのサーバのオリジンは、この文書の最後に述べる "trustworthy-origins" に含まれていなければなりません。

"reader-view-mode" 表示モードの初期値（ページめくり・横スクロール・縦スクロール）

- "paged" を指定するとページめくりモード、
- "horizontal" なら横スクロールモード、
- "vertical" なら縦スクロールモード

.....で表示を開始します。読者は随時変更できるので、あくまで最初にどのモードで表示するか、ということです。

"fix-reader-view-mode" 表示モードを固定して変更を禁止するかどうか

"yes", "no", "desktop", "mobile" の4つから選べます。

この項目の初期値は "no" で、そのままにしておくことをオススメします。

"yes" にするといつも、"desktop" ならパソコンのとき、"mobile" ならスマートフォンのときに、ユーザが表示モードを変更できなくなり、変更ボタンも表示されなくなります。コンテンツの性質によって禁止したほうがユーザ体験を向上させられるときに使います。

以降、"yes", "no", "desktop", "mobile" の4つから選択する形式の指定がいくつか登場しますが、切り替え条件はすべて同じです

"single-page-always" 2 ページの見開きにせず 1 ページで表示させるかどうか

"yes", "no", "desktop", "mobile" から選択してください。

初期値は "no" で、その場合、横書きの本をページめくりモードまたは横スクロールモードで表示したとき、画面の横幅に余裕があれば、1 画面あたり 2 ページの見開きで表示します。

"yes" にするといつも、"desktop" ならパソコンのとき、"mobile" ならスマートフォンのときに、画面の横幅が広くても常に 1 画面あたり 1 ページで表示させます。

"autostart" 貼り付け時、再生ボタンのクリックを待たずに自動的に本を開くかどうか

ビビはウェブページに埋め込んで使えますが、データ量の大きい本を自動再生すると、親ウェブページの読み込み処理を遅くしてしまう場合があります。とはいえ、軽い本なら自動再生してしまいたいかもしれませんし、パソコンとスマートフォンで挙動を切り替えたいこともあるでしょう。そこでこれも、"yes", "no", "desktop", "mobile" の値で、切り替えられるようにしました。

"yes" なら、どんなときもいつも自動再生します。"no" は逆に、必ず再生ボタンのクリック／タップを必要とします。"desktop" は、パソコンなら自動再生、スマートフォンでは再生ボタンの操作が必要になります。"mobile" は "desktop" の逆です。

"start-in-new-window" 貼り付け状態で再生ボタンを押したとき、新しいウィンドウで開くかどうか

貼り付け時、"autostart" の設定によって再生ボタンが表示された場合、その再生ボタンをクリック／タップしたとき、そのまま貼り付け状態で再生するか、新しいウィンドウで開くかどうかの設定です。これも "yes", "no", "desktop", "mobile" の4首から選べるようになっています。

"yes" はいつも別ウィンドウ、"no" はかならず埋め込み状態、"desktop" はパソコンなら別ウィンドウでスマートフォンなら埋め込み、"mobile" は "desktop" の逆、です。

ごく最近まで、ビビはいつも "mobile" と同じ挙動でした。スマートフォンでも埋め込んだまま起動したいという要望はちらほらとあったのですが、読むのに十分なパフォーマンスが出なかったためです。

が、どうやらそろそろまともに動くようになったようなので、任意で選べるようにしました。ただ、スマートフォンの場合、狭い画面内のさらに小さなフレームで開くと読者に不便を強いるかもしれません。よく気をつけて使ってみてください（ユーザは、メニュー内のボタンからいつでも別ウィンドウで開きなおせます。とはいえ）。

"use-nombre" ページ番号・パーセントインジケータを使うか

ページをめくったときなどに、ビビは現在のページと全体のうちのパーセンテージを表示します。これを使うかどうか決められます。

これも "yes", "no", "desktop", "mobile" の設定で、初期値は "yes" です。

"use-slider" ページスライダを使うか

画面の真ん中あたりをクリック／タップすると下端に出てくるページスライダの使用有無です。本の中の好きな位置にジャンプできる便利な UI で、有効になっていると、パソコンではページめくりモードのときに擬似的なスクロールバーの役割も兼ねます。

これも "yes", "no", "desktop", "mobile" の設定で、初期値は "yes" です。

"use-arrows" 画面端のクリック／タップに反応するページめくり機能を使うか

ビビは画面左右にページめくり領域を用意して、クリック／タップで操作できるようになっています。ただ、特別なプログラムを組んで埋め込まれた親ページやアプリに独自の UI をつけて不要になったりしたときは、消せるようにしてあります。

これも "yes", "no", "desktop", "mobile" の設定で、初期値は "yes" です。これがないとめくれなくなってしまう場合がほとんどなので、基本的には "yes" のままをおすすめします。

"use-keys" キーボード操作を受け付けるか

有効になっていると、キーボードのカーソルキーでもページ移動ができます。

これも "yes", "no", "desktop", "mobile" の設定で、初期値は "desktop" です。有効にしてもスマートフォンでは使えない機能なので、"desktop" か "no" をおすすめします。

"use-cookie" 状態を Cookie に保存して次回開いたときに再現するか

有効になっていると、最後に開いていたときの表示モードとページ位置を保存して、次回も同じ本は同じ状態で開始します。

本の特定には、ビビのディレクトリとEPUB のメタファイルに書かれた本の ID (identifier) の組み合わせが使われます。記録されるのは表示モードとページ位置だけで、この機能によってプライベートな情報が外部に漏れたりはしません。

これも "yes", "no", "desktop", "mobile" の設定で、初期値は "yes" です。パソコンとスマートフォンで挙動を分ける理由は特にないと思いますので、"yes" か "no" かで選ぶのがよいと思われます。

"cookie-expires" Cookie を保存するときの有効期限（単位：ミリ秒）

上の "use-cookie" が有効になっている場合に保存される Cookie の有効期限です。

単位はミリ秒（1/1000 秒）で、初期値は $1000 \times 60 \times 60 \times 24 \times 31$ 。これは $1000 \times 60 \times 60 \times 24 \times 31$ という意味で、最初の 1000 は 1/1000秒が1000なので1秒、つづけて、その60倍で1分、その60倍で1時間、その24倍で1日、その31倍なので、つまり31日間となっています（あらかじめ計算した値の 259200000 と書いておいても同じことですが、バラしたほうがわかりやすいかな、と）。

どんなに長くても1年以内にしておくのがいいと思います。たとえば読んでいる途中のリロードだけに対応したいなら数分程度でもいいのです。いろいろ試してみてください。

"ui-font-family" 目次やその他の基本 UI のフォント指定

たとえば、台湾では句読点をボックス中央に配置したフォントが一般的なのでそういうフォントを使いたい、などなどの事情があるとき、デフォルトフォントを指定できます。値は CSS の font-family そのままです。

これは本文には影響しません。あくまで、ビビの UI として表示される目次などのフォントです。

項目の説明 2：本の中身の表示カスタマイズ

"book-background" 「見開き」の下背景

「見開き」の白い箱のうしろに、暗い色のエリアがあります。そのスタイルを変更できます。値は CSS の background そのままです。つまり背景画像なども使えます。

"spread-gap" 「見開き」と「見開き」の間隔

縦／横スクロールモード表示時の、「見開き」同士の間隔です。単位は px。

最初と最後の「見開き」とウィンドウ端の距離は、この値にかかわらず、ウィンドウの真ん中に来るようにウィンドウ端との距離が調節されます。

"spread-margin" 「見開き」とウィンドウの間隔

「見開き」とウィンドウの、「幅」方向（本の進行方向と垂直の軸）の間隔です。単位は px。

"spread-border-radius" 「見開き」のカド丸設定

「見開き」の白い箱のカドを丸めることができます。CSS の border-radius に適用する値そのままです。

"spread-box-shadow" 「見開き」に影を付けるなら CSS を

「見開き」の白い箱に影をつけられます。CSS の box-shadow に適用する値そのままです。

影を付けるとすこしスクロールがもたつくかもしれませんので、ないほうがいいのかもかもしれません。不要なら "none" もしくは "" としてください。

"item-padding-left" / "-right" / "-top" / "****-bottom" 各アイテム内の余白

「見開き」の内側、EPUB 内の各 HTML が描画される領域までの余白です。数値の単位はすべて px（ピクセル）。

グチですが.....ホントは全部 0 でいいはずじゃないかと思うんです。EPUB 内の CSS で作者が制御できるなら、その方がいいはずだから。でも、世間には制御不能な余白をつけるリーディングシステムも多く、そうしたリーディングシステムに合わせて、余白を設定していないか狭めにしている EPUB はかなり多いです。この項目はそのためにあります。

EPUB 内の HTML に適用された背景色・背景画像もこの余白に反映されるように処理しています。ただし、本が固定レイアウトのときは無視され、ビビによる余白調整は行われません。

項目の説明 3：エクステンション

ビビには、連携プログラムを追加して機能を拡張できる仕組みが用意されています。こういう仕組みは一般にプラグイン、エクステンション、機能拡張などといわれますが、ビビではエクステンションと呼ぶことにしました（ビビは女の子なので、アクセサリっぽいのがいいでしょう？）。

エクステンションの基本

たとえば、example という名前のあたらしいエクステンションを手に入れたとします。そのエクステンションが example/script.js というフォルダ構成になっていれば、まず、example フォルダを bibi/extensions/ フォルダに入れます。

次に、プリセットファイル内の "extensions" で始まる行の次の行に、

```
{ "src": "../extensions/example/script.js" },
```

.....と書き足します。"src" に指定するのは、そのプリセットファイルからの相対パスです（bookshelf ディレクトリの項目で説明したのと同じように、bibi/extensions 以外のフォルダに置いたファイルや、許可されていれば別サーバを指定することもできます）

すると、ビビは起動時にそのエクステンションを追加で読み込み、エクステンションの機能が有効になります。

配布時にいくつか同梱されていますが、これらはどれも、必要に応じて自動で読み込まれるか、必要だと思う人が自分で有効化すればよいようにデフォルトでは読み込まないようにしてあるか、どちらかです。以

下、同梱エクステンションについて説明します。

"Unzipper" 展開していない (= Zip アーカイブ済みの) EPUB を扱えるようにする

必要に応じて自動的に読み込まれる特殊なエクステンションです。

EPUB は ZIP フォーマットでアーカイブされており、ブラウザはふつうは ZIP アーカイブの内部にあるファイルを扱えませんが、このエクステンションによって、ビビはアーカイブ状態の EPUB も扱えるようになります。

受信したパソコン／スマートフォンでビビが展開して処理する必要があるため、EPUB のファイルサイズが 10MB を超えるあたりから、読み始められるまでの処理時間の長さが顕著になってきます。そもそも重くて転送時間のかかるファイルを受信した上でさらに展開処理が発生するので、仕方のないことです。また、古くて非力なスマートフォンなどでは、展開の処理負荷で表示以前にビビが止まってしまうこともあります。

ただし最近の改善によって、軽い本の場合は、展開済みの各ファイルをバラバラにダウンロードするよりも読み込みが効率化されて、高速・低負荷になる場合もあります。一度表示してしまえば動作速度に大きな差はありません。

ちょっと前まで、ビビを使って本を公開するときには、あらかじめ展開しておくことをおすすめしていました。が、Internet Explorer でも展開できるようになり、全般的に速度も改善し、場合によっては Zip のままのほうが速いこともあるので、「予め展開しておくほうが速い場合が多いですよ」程度の推奨に緩めることにします。

数年前のビビ開発初期はむしろアーカイブ状態のままでしか扱えなかったくらいビビの基本的な機能で、また、圧縮ファイルをドラッグ&ドロップで開けるのも、自作 EPUB のプレビューなどに便利なビビのメリットです。

"Analytics" Google Analytics でユーザの行動を分析できるようになる

本をウェブで公開するとなると、どれだけの人が再生ボタンをクリックしたか、目次のどの項目をクリックされたか、最後まで読んだか、などなど、気になってくると思います。これらを、Google Analytics のイベントトラッキング機能を使って記録できるようにします。

Google Analytics の基本機能として同時に記録される OS や アクセス元言語圏などなども合わせて、読まれた状況をさまざまに分析できるようになります。

有効化するには、"src" に加えて "tracking-id" という値で Google Analytics のトラッキング ID (書式 : UA-XXXXXXXX-X) を書き足す必要があります。

現在記録されるのは、下に挙げるユーザ行動です。ほかに記録・分析したいことがあれば、作者にご要望をお寄せください。

- 再生ボタンのクリック／タップ
- 目次項目のクリック／タップ
- 最後のページの表示

最近、ビビを使ったお仕事の中で、非常にたくさんの情報を記録する独自拡張を開発しました。そのクライアントのご厚意で、処理自体はオープンソース化のご許可をいただいていますので、そのうち、大きく拡張するかもしれません。

"EPUBCFI" EPUBCFI 仕様の位置・範囲指定を利用できるようにする

EPUBCFI は、複数ファイルで構成される EPUB 内の、特定ファイルの特定段落の何文字目から何文字目、といった位置・範囲を指定することができる仕組みです。「これこれについてはこの本のここにかいてあります」といったリンクで人に伝えたりしやすくなります。論文の引用などにも使えますね。

EPUB ビューワは EPUBCFI に対応していることが望ましい、ということになっており、実際うまくつかえば便利なのでビビも部分的に対応してみたのですが.....書式が「人間の読み書きできるもんじゃねえ」という感じだからか、EPUBCFI 仕様自体があまりつかわれていません。このプラグインも、仕様に用意された機能の完全対応は諦めて、位置の指定のみに対応して開発が止まっています（範囲指定はごく部分的にしか扱えません）。

ビビには、EPUBCFI を使わなくても本の途中を指定して開く機能を別につけてあります。それは EPUB の基本仕様ではなく独自機能ですが、もうすこしわかりやすく直感的な書式で指定できるようになっています。別途解説予定ですので、おたのしみに（といってぜんぜん解説しないままですみません）

項目の説明 4：高度な設定（普通の使い方をするかぎり変更しないことをおすすめします）

"trustworthy-origins" bookshelf ディレクトリの場所や Message の送信元として許可するオリジン

必要であれば外部のオリジンを列挙することで、bookshelf を外部サーバに置いたり、別サーバからの Message で動作指示を受け取ったりできるようになります。なにも書かなくても、ビビ自体が置かれたサーバのオリジンは許可されます。

たとえばここに "https://dl.dropboxusercontent.com" を加えれば、自分のウェブサーバに設置したビビから Dropbox の公開フォルダに置いた EPUB を読み込んだりできるようになります。ただし、このオリジンはどんな Dropbox ユーザの公開フォルダの URL でも共通なので、まったく知らない誰かが、その人の Dropbox 公開フォルダに置いた EPUB を、あなたのビビで、表示させることもできるようになります。

十分に注意して、ドキュメントルート以下すべてが自分の管理下にあるサーバのみを加えるのがいいと思います（その条件では、Dropbox も除外されます）。

もちろん、その外部サーバがビビの置かれたサーバまたはすべての外部サーバからのリクエストを許可していなければ、ここに加えても読み込むことはできません。