簡易マークアップとAH Formatterによる商業品質を満たした技術書籍制作フローの展望

株式会社トップスタジオ
武藤 健志
@kmuto

先に結論から

AHFormatter ＋ CSS組版

はまだ本格導入には難あり

（※当社で制作している書籍において）

でも

未来

は感じる

お話すること

- 当社の紹介
- Re:VIEW+InDesign DTPとその制約
- Re:VIEW+AH Formatterの状況と展望

(株)トップスタジオ

- 「編集プロダクション」
- 出版社（版元）が顧客のBtoB事業
- メーカー等のカタログ、マニュアル、ガイドブック等も制作
- 「印刷所の手前まで」を請け負う、総合プロダクション
- 企画、翻訳、執筆、編集、検証、校正、組版、カバー/紙面デザイン、電子書籍化

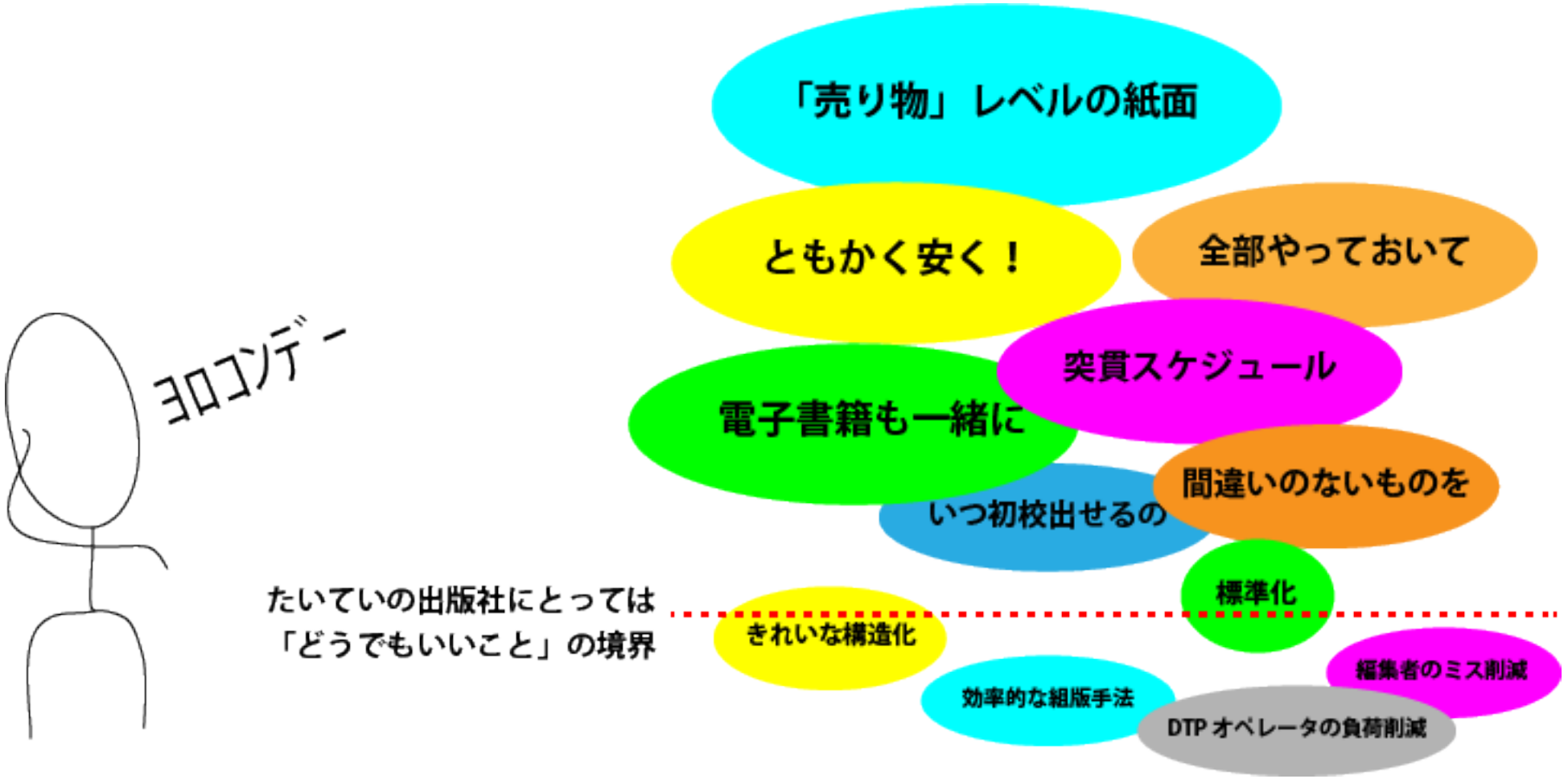
http://www.topstudio.co.jp/

(株)トップスタジオ

- IT技術書
  - 技術書籍出版社の取引先多数
  - ソフトウェア、ハードウェア、プログラミング、ネットワーク、ソーシャル、モバイル、etc...
- 資格書
  - IT技術資格、福祉資格、通関士等
- カタログなど
  - Cisco社さま製品カタログ、大手メーカーマニュアル、英語辞書、文芸書 etc...
- BtoBではあるが、制作物はConsumerの目に触れることになる

(株)トップスタジオ

- 顧客である出版社の意向が最重要



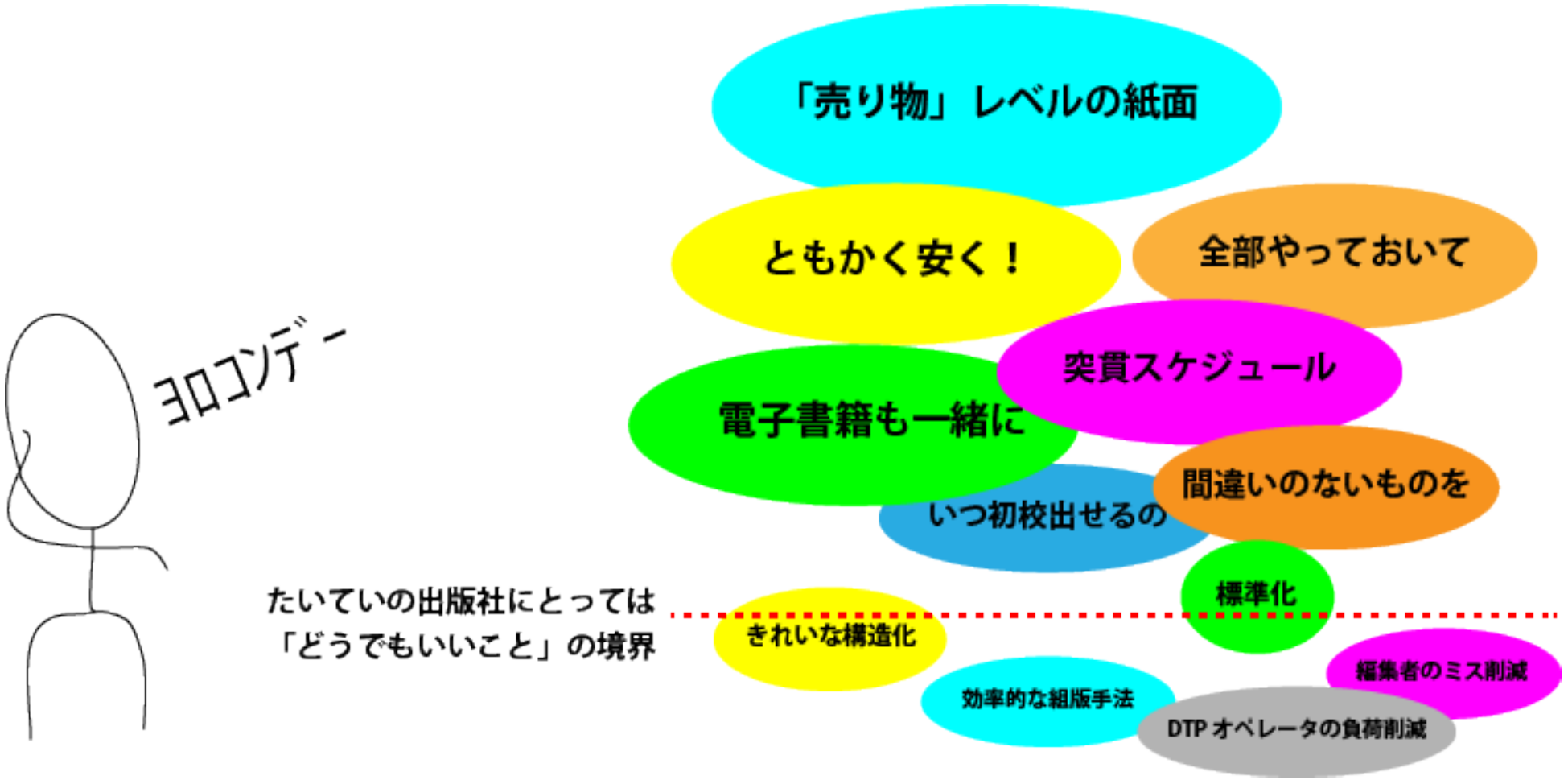
自己紹介

- 武藤 健志　(MUTO Kenshi、@kmuto)
- 株式会社トップスタジオ
  執行役員（えらくないです）
- 編集、自動組版設計、
  プログラミング その他萬事
- 前職はSGML/XML処理開発

- Debian Project公式開発者
- 執筆、監修、翻訳
- TeX好き、ものぐさ

再掲

- 顧客である出版社の意向が最重要ではあるが...



ミッション

- 出版社の意向に応えるには...

制作手法の改善が必要

- InDesignファイルのようなバイナリファイルをマスターデータにしない
  - DTPオペレータしか触われない代物に
  - EPUB化はちょっと凝ったデザインだけで破綻
- 原稿データを常に更新し、文責のある人（編集者 or 著者 or 訳者）が直接修正できる
- 原稿データから変換して最終的に組版（PDF）が容易にできる

制作手法の改善が必要

- ただし、
  - 紙面デザイン、要素の配置方法は本ごとに変わる
  - シリーズ書でも微妙に違ってたり

- 汎用性があり、かつカスタマイズできる制作フローが必要

適切な原稿フォーマット

- では、原稿フォーマットに適切なのは何か？（※技術書籍において。）
- 独断と偏見とポジショントークです
- XML (InDesignに利用できる)、HTML (EPUB用)、テキスト（どうにもならなくなったときに従来のやり方で組版できる）に簡単に変換可能であること
- 特定の紙面デザインに限定されないこと

適切な原稿フォーマット – Word?

- スタンダード
- 構造化とは無縁
  - スタイル付けなどすごいがんばれば…
    でもそれちゃんと守れる？
- 組版の元データにするには厄介事が多い
  - 原稿を差し替えて組み直すのは非現実的

適切な原稿フォーマット – XML?

- 構造化文書の王道
- 拡張しやすい
- マークアップ（タグ）がたくさんで
  読みにくい、編集しにくい
- メタ情報が区別しづらい
  - 紙面には出さないが原稿には入れておきたい情報。
    コメントや作業連絡など
- 文字のエスケープが頻出する
  - <→<、>→>、&→&

XMLがhuman readable?

『XML & InDesign』（2013, オライリー・ジャパン）

適切な原稿フォーマット – HTML?

- XMLより覚えることは少ない（マークアップの種類が少ない）
- Webブラウザで確認できる
- 拡張性は低い
- HTMLだからといって、紙とEPUBでの併用はあまり現実的でない
- メタ情報を入れにくい
- 章節項番号や図表番号等の採番機能がない
- HTMLBook?

アンテナハウスさんのアピールとは相反するかも...

適切な原稿フォーマット – その他

- TeX
  - 完全バッチ型、数式表現や採番処理など優れた機能
  - 紙面化を含んだフローであり、その紙面デザインおよびスタイル作成に高度な技術を要する
  - XML化やプレインテキスト化は難
- Markdown、Wiki
  - 単純なマークアップ
  - githubの人気とともに最近広まりつつある
  - 相互参照や採番機能の不足、マークアップと地の文の区別がしづらい

    ※pandocはまぁまぁいいかも

適切な原稿フォーマットがない...

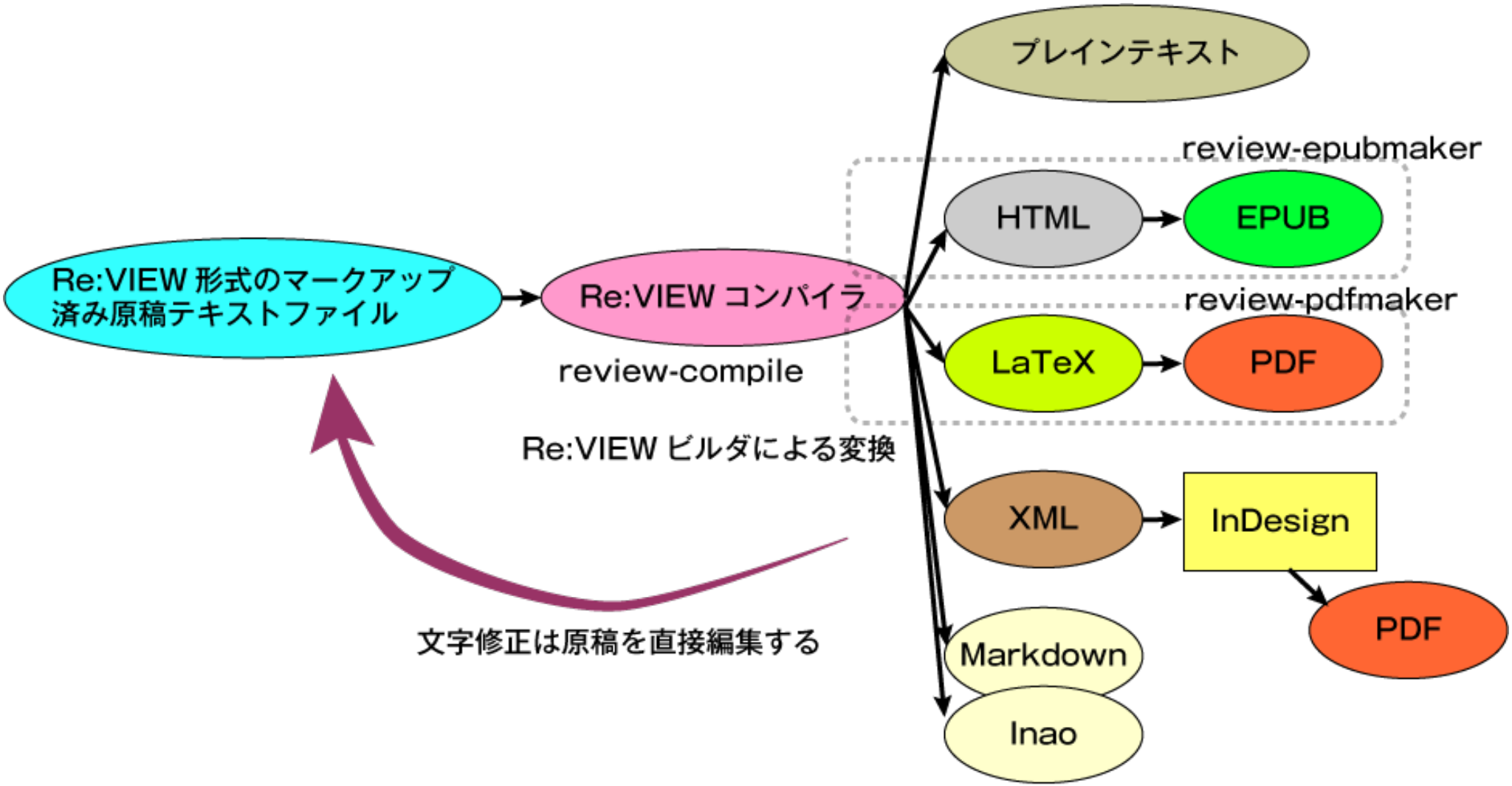
結局、マスターデータとして使い続けられるような原稿形式がなかった

- 2007年当時。
- 今だとpandocがわりと良いように思える

Re:VIEW（れびゅー）

- 私が書籍の編集を担当した青木峰郎さんが、ご自身の執筆環境としてRe:VIEWの原型を開発
- 簡易なマークアップながら、相互参照や自動採番の機能を備え、構造化されたHTMLや、テキスト、TeXなどに変換できる
- 私が開発を引き継ぎ、いろいろな書籍で利用できるように汎用化、InDesignでの利用やEPUB生成機能など
- 達人出版会、オライリー・ジャパン等での標準形式の1つに

Re:VIEW



https://github.com/kmuto/review

Re:VIEW原稿

ch02.re

CR/LF　Unicode（UTF-8）　ReVIEW
行末コード　エンコーディング　カラーリング　情報を

```
= スプライトを作ろう

ゲームの世界では本来、アセットパイプラインとは開発しているゲームのビジュアルなワークフローを指す。マニュアルでゲームのメディアフォルダにファイルをコピーするだけの単純明快なものから、複雑な自動化スクリプトを書いてアートワークを生成するものまで、さまざまある。Impactを使った開発では、ゲーム内で使用されるグラフィックはすべて、スプライト（sprite）として扱う。

== スプライトとスプライトシート

Impactでは、アートワークの表示とアニメーションは「スプライト」という形式で行う。スプライトは、ディスプレイに表示されるシングルビットマップイメージで、この場合、HTML5 Canvas要素になる。関連するスプライトは、整理しやすいように、スプライトシートと呼ばれる1つの画像にまとめることができる。

//image[Figure2-1][これから作成するゲームのスプライトシートの例]{
//}

@<img>{Figure2-1}は、メインキャラクターの動きに用いるすべてのビジュアルを含むスプライトシートだ。スプライトシートは通数の組み合わせで構成される。この例では、160×16ピクセルの大きさのスプライトシートに、それぞれ16×16ピクセルのスプライトシートの長さを16で割れば、スプライトが10個あることがわかる。アニメーションにスプライトシートを使う場合には、ゲームエどのアニメーションセットの一部であるかを指示する。Impactの例を下記に示す。

//emlist{
this.addAnim( 'idle', 1, [0] );
this.addAnim( 'run', 0.07, [0,1,2,3,4,5] );
this.addAnim( 'jump', 1, [9] );
this.addAnim( 'fall', 0.4, [6,7] );
//}
```

行: 1 / 119　文字: 0 / 4,888 (>0)

『初めてのHTML5ゲームプログラミング』（2012, オライリー・ジャパン）

EPUB変換

Adobe - Adobe Flash Player
Readium
chrome-extension://fepbnnnkkadjhjahcafoaglimekefifl/index.html?epub=filesystem%3Achrome-...
Readium

2章

スプライトを作ろう

ゲームの世界では本来、アセットパイプラインとは開発しているゲームのビジュアルなワークフローを指す。マニュアルでゲームのメディアフォルダにファイルをコピーするだけの単純明快なものから、複雑な自動化スクリプトを書いてアートワークを生成するものまで、さまざまある。Impactを使った開発では、ゲーム内で使用されるグラフィックはすべて、スプライト（sprite）として扱う。

2.1　スプライトとスプライトシート

Impactでは、アートワークの表示とアニメーションは「スプライト」という形式で行う。スプライトは、ディスプレイに表示されるシングルビットマップイメージで、この場合、HTML5 Canvas要素になる。関連するスプライトは、整理しやすいように、スプライトシートと呼ばれる1つの画像にまとめることができる。

図2-1　これから作成するゲームのスプライトシートの例

図2-1は、メインキャラクターの動きに用いるすべてのビジュアルを含むスプライトシートだ。スプライトシートは通常、計算しやすいサイズやアイテム数の組み合わせで構成される。この例では、160×16ピクセルの大きさのスプライトシートに、それぞれ16×16ピクセルのスプライトが並んでいる。つまり、スプライトシートの長さを16で割れば、スプライトが10個あることがわかる。アニメーションにスプライトシートを使う場合には、ゲームエンジンに対して、どのスプライトがどのアニメーションセットの一部であるかを指示する。Impactの例を下記に示す。

Chrome+Readiumで表示

Re:VIEW+InDesign自動組版

①Re:VIEW原稿から汎用XMLに変換

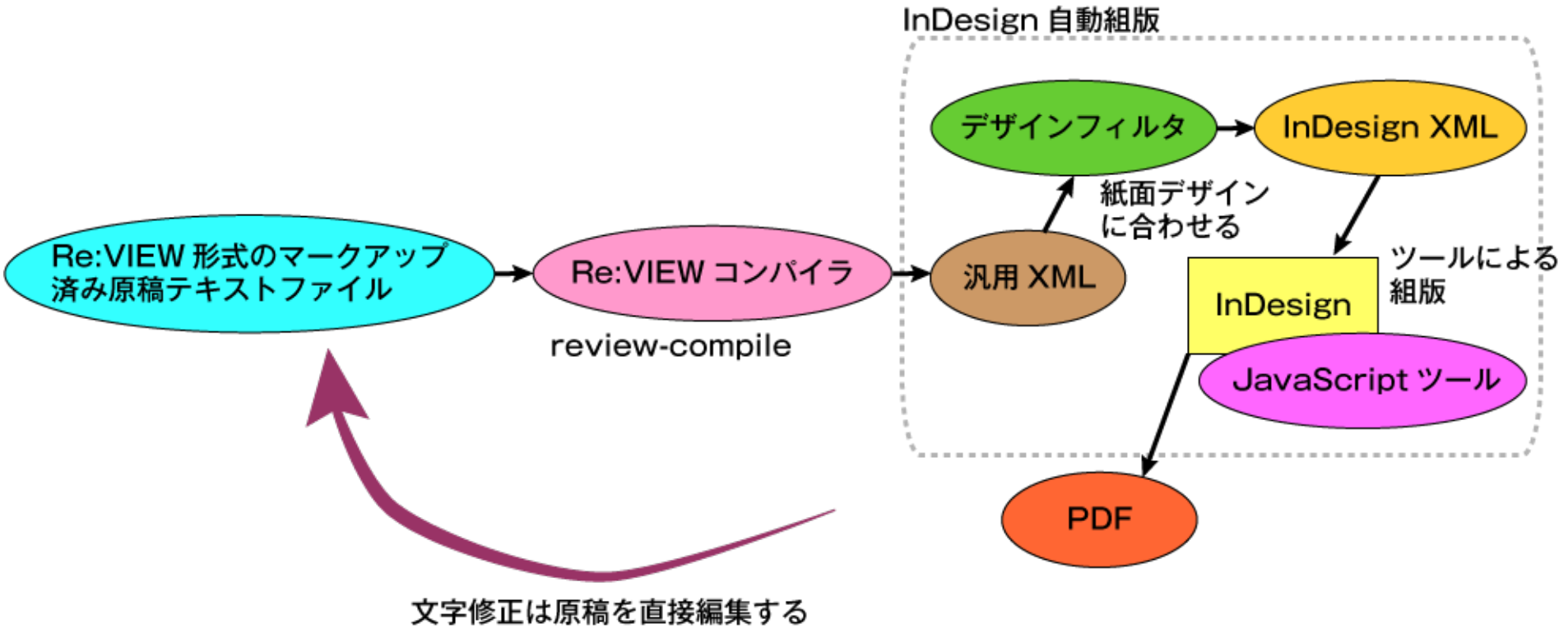
②汎用XMLを書籍の紙面デザインに合わせた加工フィルタに通し、InDesign XMLへ

③InDesignの紙面テンプレートに投入し、手での組版を近似する手法の独自のJavaScriptで紙面作成

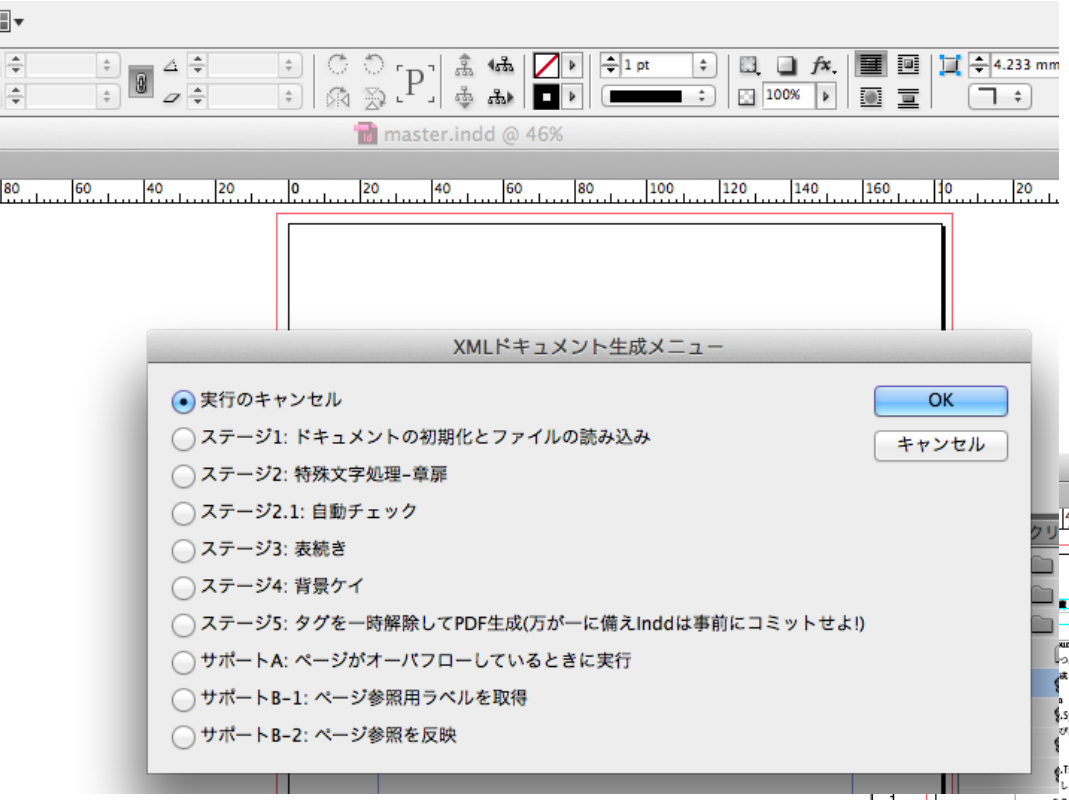
④文字修正はRe:VIEW原稿を編集し、必要なら再度InDesignで紙面作成

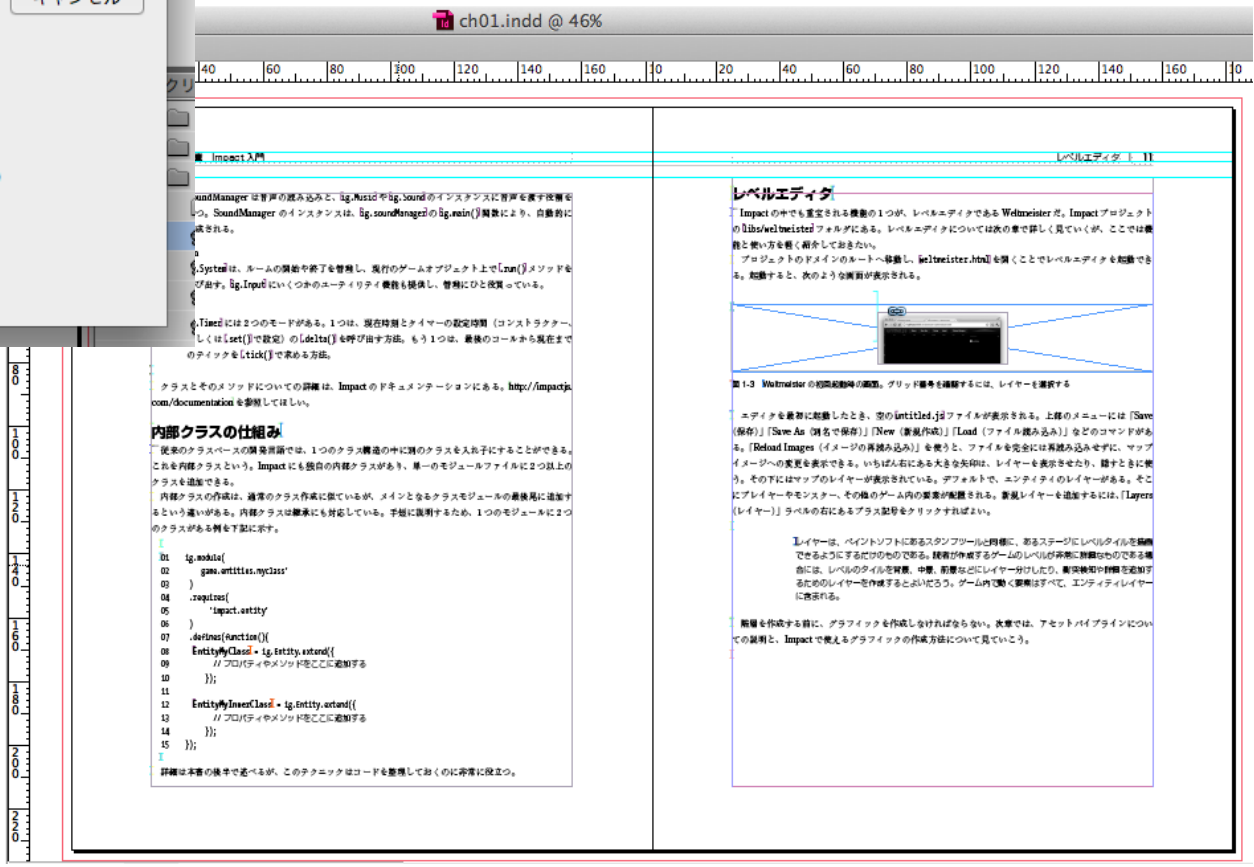
- http://www.topstudio.co.jp/　に近日ムービー掲載予定

Re:VIEW+InDesign自動組版



Re:VIEW+InDesign自動組版



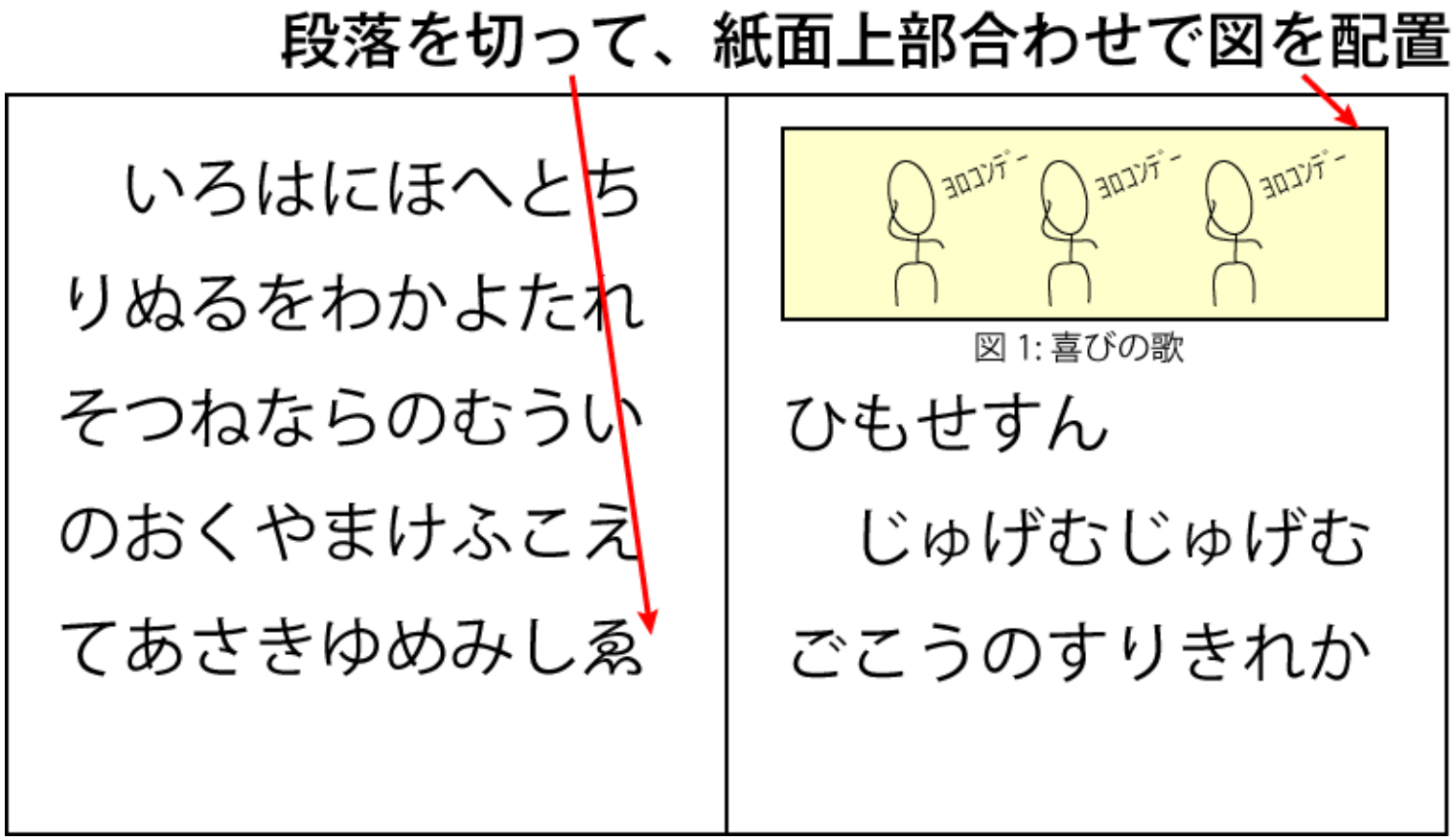


InDesign自動組での表現の課題

- 50冊以上作ってきた結果
  - オライリー・ジャパンさまのシリーズ書のほか、各社の千差万別な紙面
  - おおむね満足いただけている模様
- どうにもならない制約も見えてきた
  - あきらめて「手でがんばる」ならなんとでもなるが…

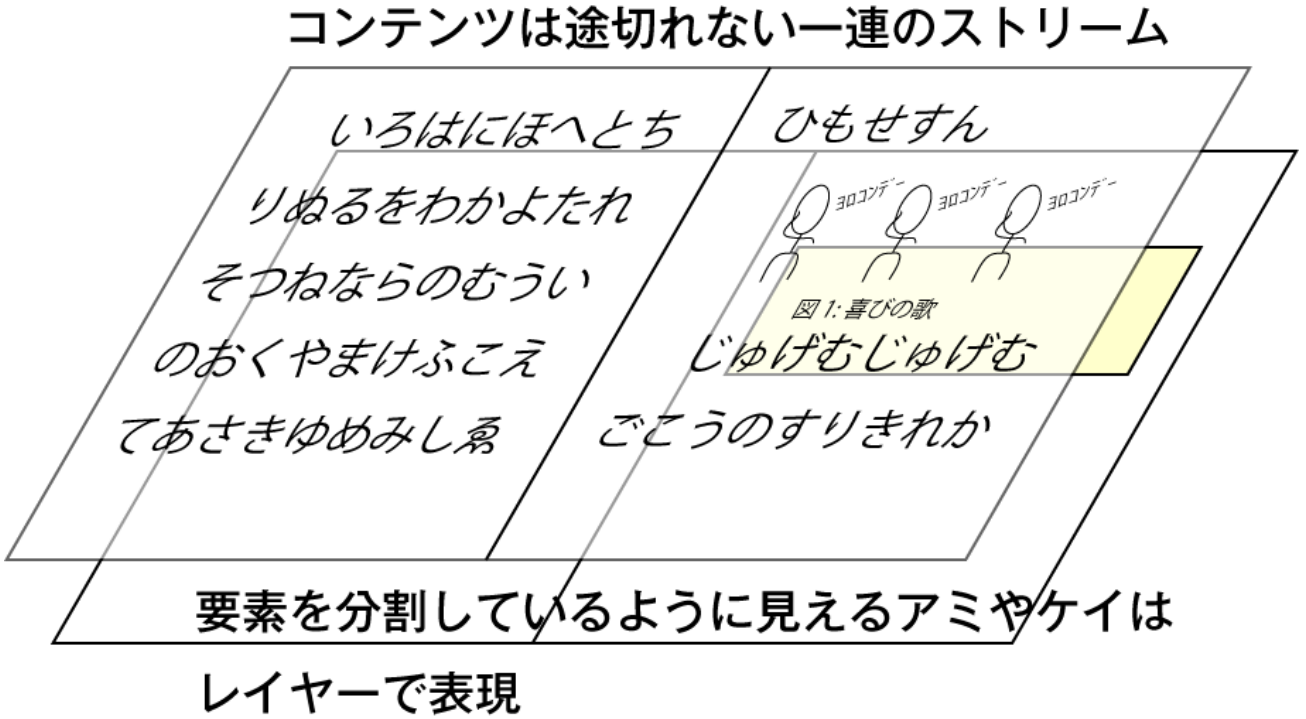
InD課題1: フロートの扱い

- 「図版を本文から独立して版面の上（あるいは下）に合わせたい」



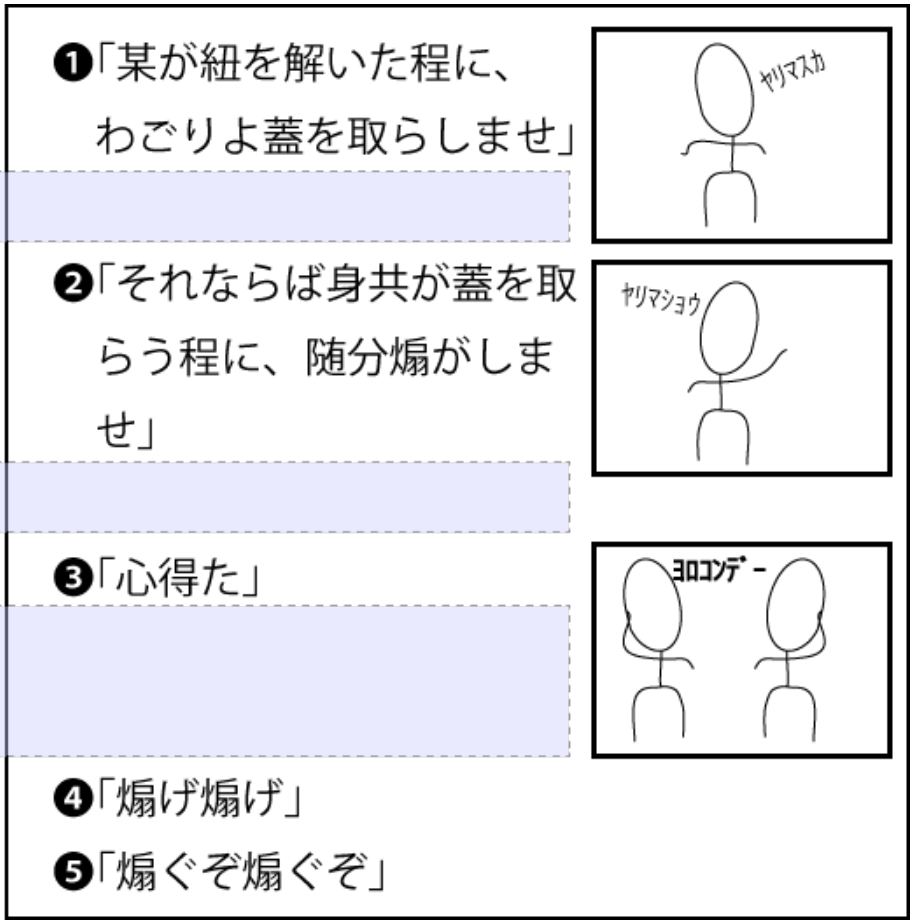
InD課題1: フロートの扱い

- InDesign自動組版は、コンテンツが一連のストリームになっていることが前提
  - フロートに見える囲みなどはレイヤー機能
  - ストリームからの切り出しは配置が困難



InD課題2: サイドバイサイド

- 「手順で左に番号と説明、右に図を入れて、関係が明示的にわかるようにしたい」



InD課題2: サイドバイサイド

- 課題１と同様の理由で、一連のストリームにする必要がある
  - 「グルーピングしてインラインテキストフレームとして配置する」手動組版の王道が使えない
- 実際に配置しないと、箇条書き間にどの程度の空白行が必要かわからない
- 実際に配置してから迂闊にJavaScriptでいじるとクラッシュする

InD課題3: バーティカルリズム

- 「見開きで左右のページの段落ベースラインが揃うようにしたい」

ベースラインのリズムに合わない要素。上下アキを適切に調整して後続をリズムに戻すことが望ましい

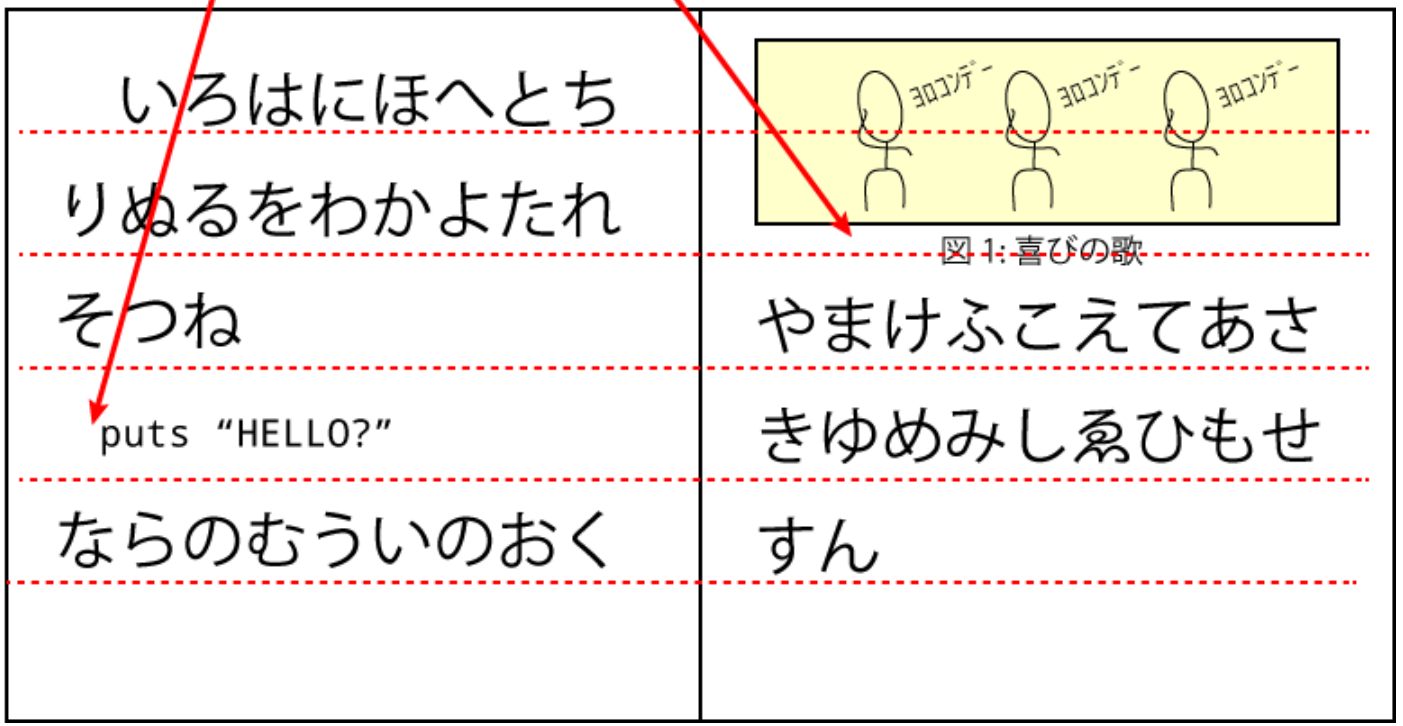


図 1: 喜びの歌

InD課題3: バーティカルリズム

- 「配置してから自動で調整」は困難（不安定化）、「配置してから手で調整」はきりがない
- プログラムコードや図表など、標準のベースラインと合わないサイズのものが多発すると、揃える労力が多大
  - 手動組版だと、そのような場面のたびに必死に調整していることが…

AH Formatterによる解(1)

- InD課題1: フロート
  - float、-ah-float-move、-ah-float-x、-ah-float-yといったCSS属性で版面の上や下への合わせが可能
  - ただし、-ah-float-moveで無茶するとクラッシュ
  - top/bottom等は紙面の見栄えを見て恣意的に人間が判断することに変わりはない。

    ※HTMLの指定要素のclassパラメータを調整することになるので、手間はかかる

AH Formatterによる解(2)

- InD課題2: サイドバイサイド
  - 課題1と同じフロートで対処できる
  - ブロック化してスタイル付け はHTML+CSSの得意分野

AH Formatterによる解(3)

- InD課題3: バーティカルリズム
  - column-fill属性など
  - 標準で段落以外の要素間アキを自動調整して
    おおむね「いい具合」になっているように見える
  - まだ向上の余地ありというアンテナハウス村上さん
    のtweetがあった

Re:VIEW+AH Formatter

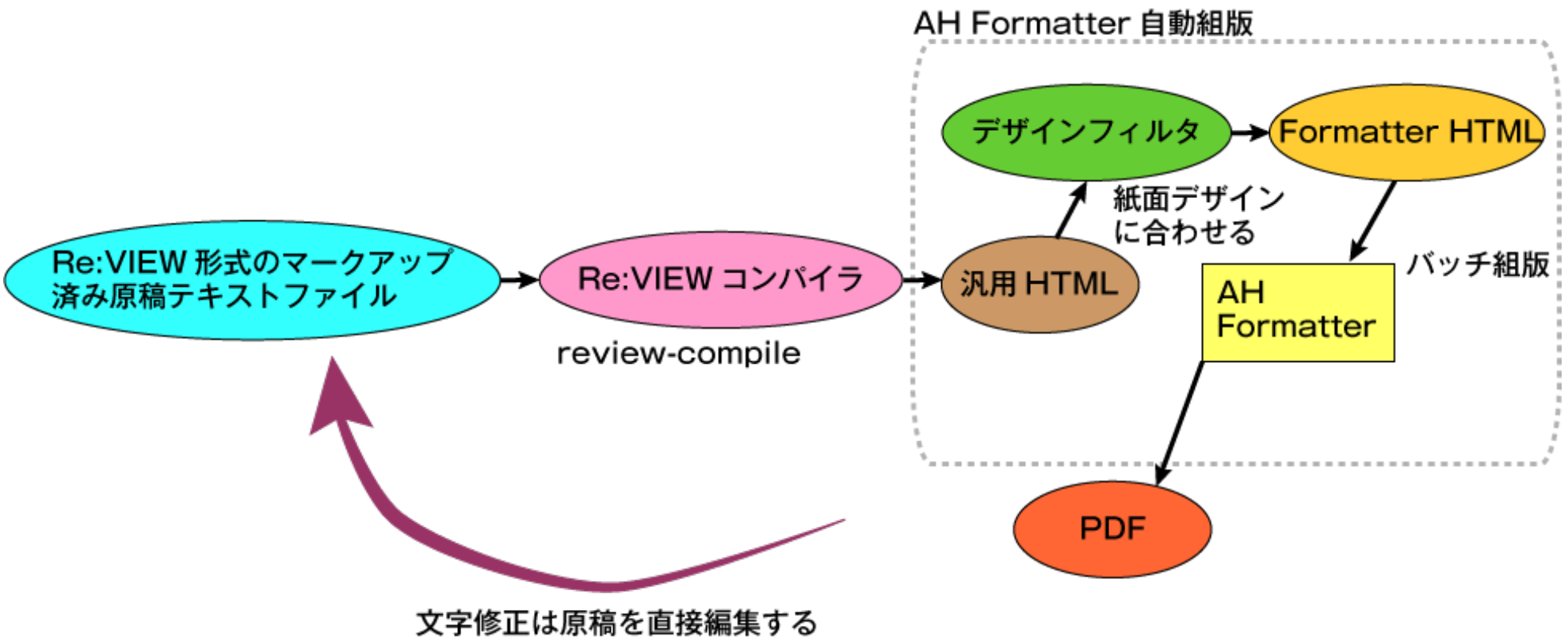
①Re:VIEW形式の原稿をマスターデータとするのは同じ

②Formatter用の汎用HTMLに変換

③紙面に合わせたHTML調整フィルタに通す

④Formatterで組版。修正があれば再びRe:VIEW原稿を編集

Re:VIEW+AH Formatter



AH FormatterのためのHTML

- Re:VIEW標準のEPUB共通汎用HTMLではAH Formatterには不都合
  - 要素に付いているid属性の値を書籍全体で固有に

    <h2 id="h1-2">→<h2 id="pre1-h1-2">
  - 脚注をインライン化

    ... 脚注内容 ...
  - 表見出し行に<thead>追加

    <table>
      <thead><tr><th>見出し</th><th>見出し</th></tr></thead>
      <tbody><tr>...</tr></tbody>

組版例

きさのスプライトシートに、それぞれ16×16ピクセルのスプライトが並んでいる。つまり、スプライトシートの長さを16で割れば、スプライトが10個あることがわかる。アニメーションにスプライトシートを使う場合には、ゲームエンジンに対して、どのスプライトがどのアニメーションセットの一部であるかを指示する。Impactの例を下記に示す。

```
this.addAnim( 'idle', 1, [0] );
this.addAnim( 'run', 0.07, [0,1,2,3,4,5] );
this.addAnim( 'jump', 1, [9] );
this.addAnim( 'fall', 0.4, [6,7] );
```

この例でわかるように、スプライト0が停止状態で、0~5が動いているアニメーションだ。スプライトを使ったアニメーションの設定については、ゲームの設定を行うところで詳述する。

ここで注意しておきたいのは、JavaScriptの配列はゼロを基点とすることだ。一番目のスプライトは必ず0で、スプライトごとに値を1ずつ増やしていく。

スプライトシートは、階層のタイル画像などのように、アニメ化されていないグラフィックにも便利だ。

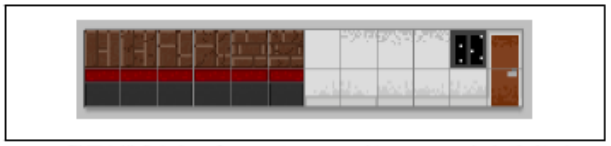


図2-2 階層を構成するスプライトはTileset（タイルセット）と呼ばれる

マニュアルでアニメーションを登録する代わりに、どのタイルが、設計している階層の壁、装飾、その他のアートワークであるかを、ゲームエンジンに指示すればよい。

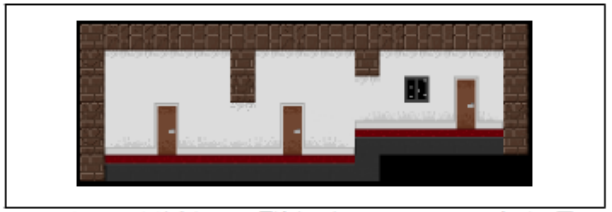


図2-3 ゲームにおけるタイルの用途を示すImpactのレベルエディタの画面

スプライトシートの基本をおさえたので、ゲーム開発において、これを実際に作成する方法について見ていこう。

ピクセルベースのゲームを開発する際にスプライトを作成する場合、すばらしいイメージエディタがいくつかある。本書ではPhotoshopを使用した例を示すが、フリーソフトではGrafX2（http://code.google.com/p/grafx2/）とGIMP（http://www.gimp.org/）を試すことをおすすめする。

Photoshopのスクリプトを使う

Photoshop上で自動化スクリプトを作る方法はいくつかあるが、ここではJavaScriptを使った例に特化したい。基本的には、PSDのレイヤ間をループし、スプライトを列に整理して単一ファイルに出力するスクリプトを作成する。ゲームにおけるグラフィック生成で最も重要なポイントは、正し

『初めてのHTML5ゲームプログラミング』（2012, オライリー・ジャパン）

CSSの管理

- AH Formatter向けのCSS管理が煩雑になりやすい
  - 紙面のためのいくぶん複雑な構造
  - さまざまな書体や級数（文字サイズ）設定
  - AH Formatter独自のCSS属性が多いため、現時点ではテキストエディタで書くしかない

CSSの管理

- Compassを導入（Rubyツール）
  - CSS上位互換のSASSあるいはSCSSで、変数やネスト、各種プラグインを使ってCSSを記述
  - Compassツールで標準のCSSに変換
- AH Formatter用の支援プラグインがあるとよいかも

AHFormatterでの表現の課題

- 私の勉強不足なだけかも
- 「案件ごとに紙面デザインがばらばらなので作り込んでいられない」という背景

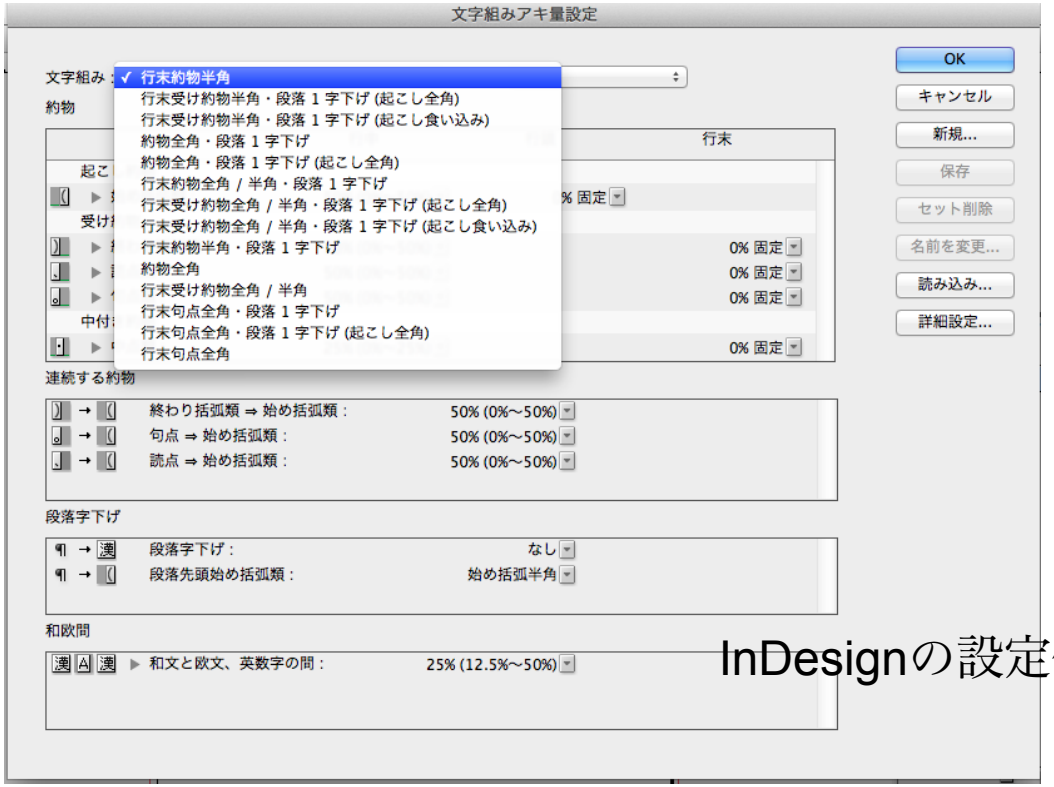
AHF課題1: 文字組の設定

- 段落文字組み、禁則、和欧文アキ量設定などパラメータは相当数あるが...

  -ah-punctuation-spacing, punctuation-trim, -ah-append-non-end-of-line-characters, -ah-append-non-starter-characters, -ah-except-non-end-of-line-characters, -ah-except-non-starter-characters, hanging-punctuation, -ah-auto-letter-spacing, -ah-avoid-window-words, -ah-justify-nbsp, -ah-kerning-mode, -ah-letter-spacing-side, text-autospace, -ah-text-autospace-width, text-justify-trim

AHF課題1: 文字組の設定

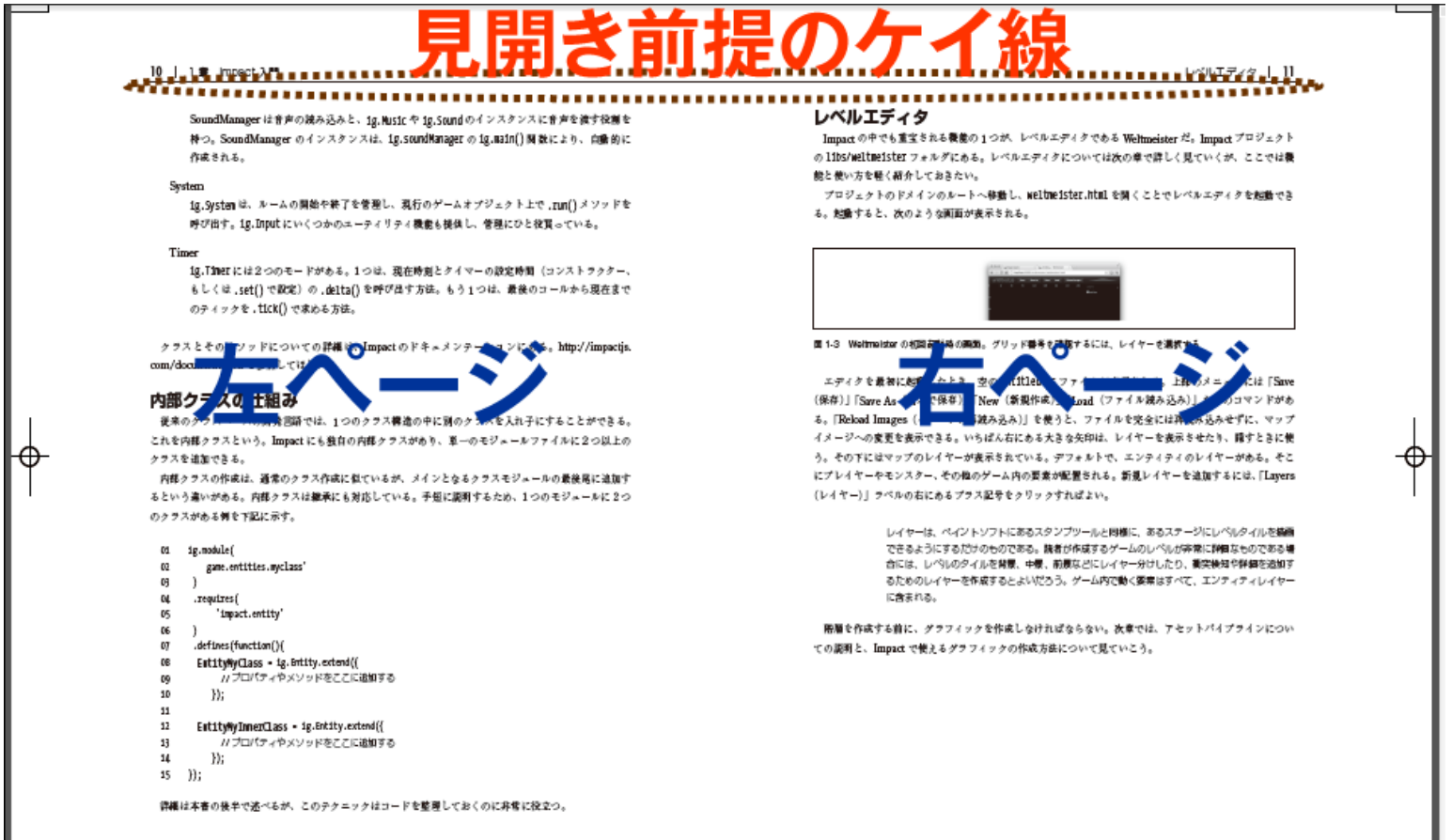
- プリセットがないので設定が大変
- InDesignの設定に比べて柔軟性に欠ける



InDesignの設定例

AHF課題2: 見開きの概念

- 左ページ、右ページの概念はあるが、見開き（スプレッド）の考慮がない
- 製本ならではの概念



AHF課題2: 見開きの概念

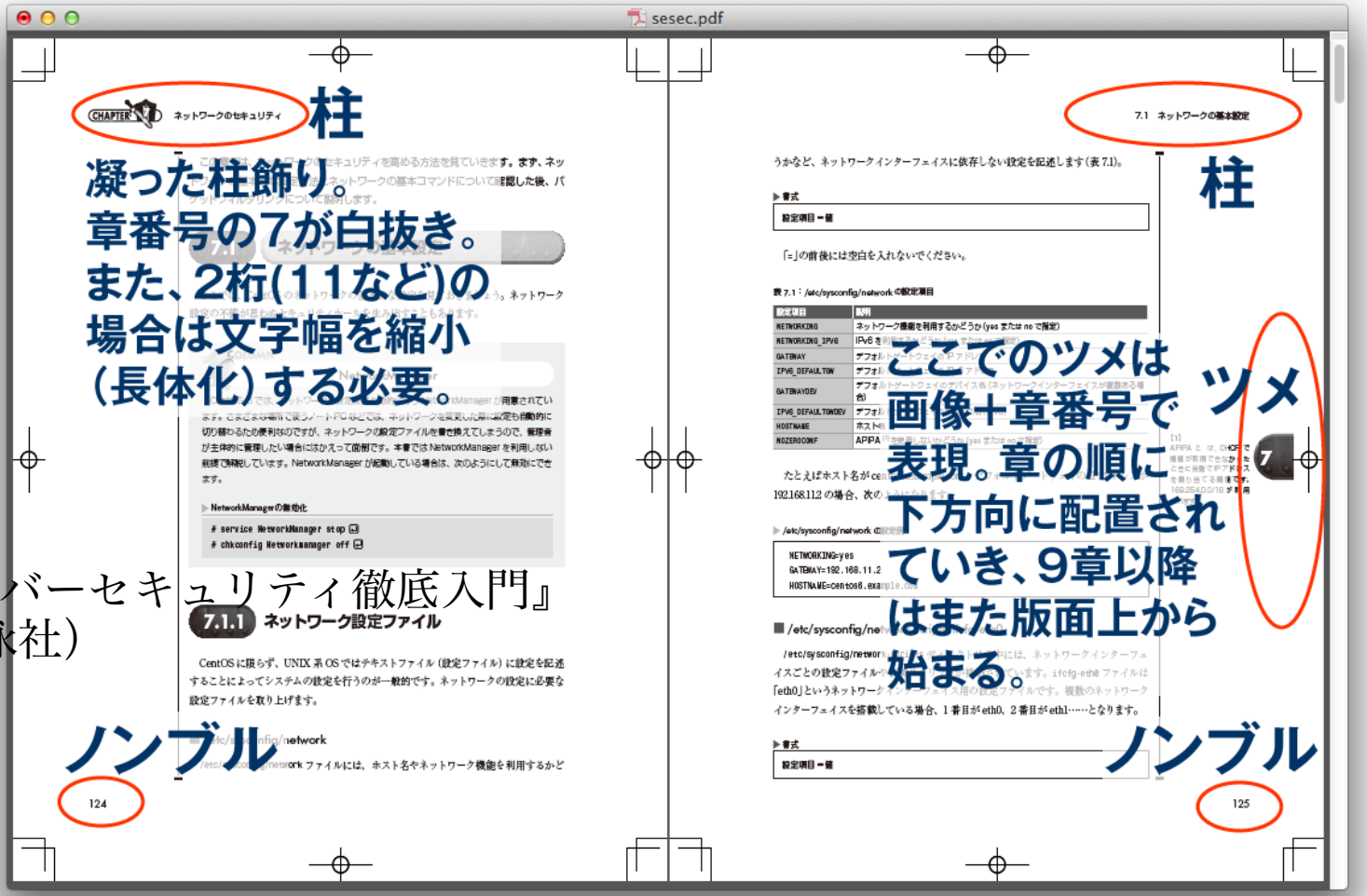
- Bad knowhow...
  - 「紙面」背景に絶対位置指定の画像で配置

```
@page Chapter: right {
  Background-repeat: no-repeat;
  Background-image: url(line.svg);
  Background-position: 0mm 16mm;
}
```

- 紙面デザイナーや出版社にとっては見開きをもって1つのデザインと捉えることがある
- 関連して「章の最後のページ」も判断できない

AHF課題3: 版面外の要素調整

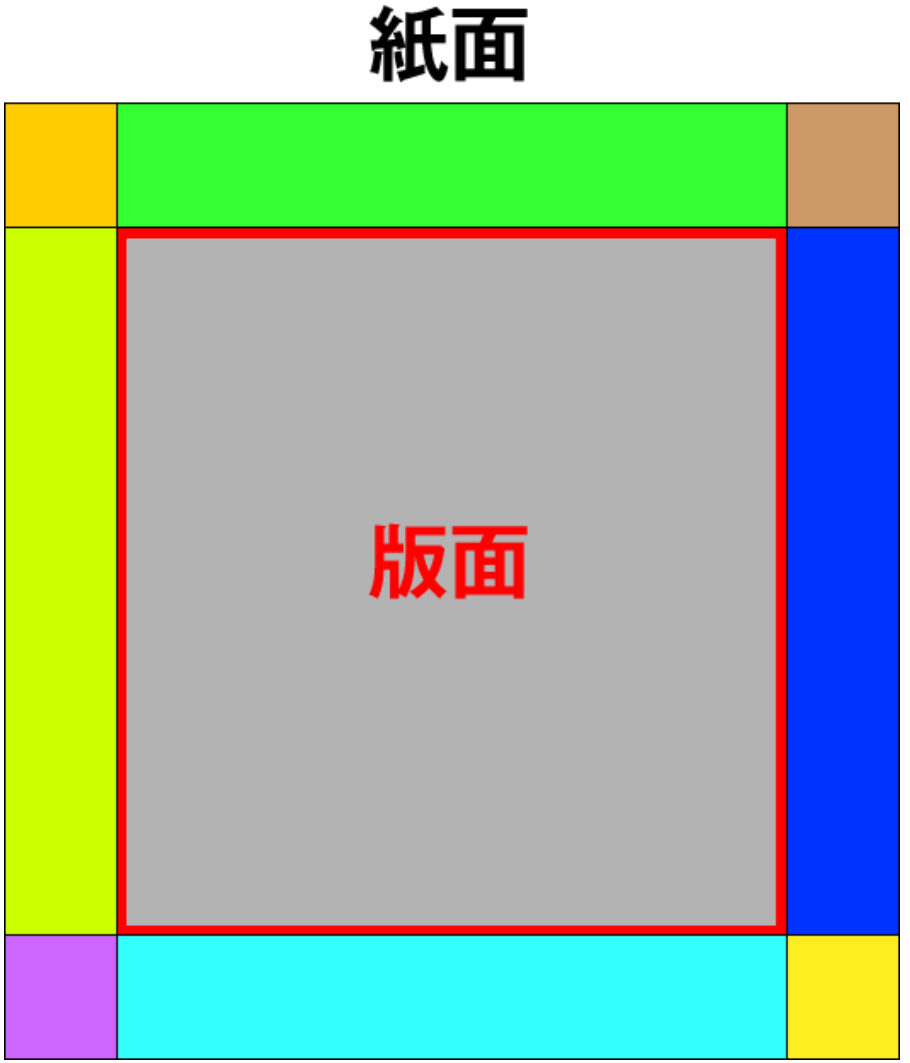
- 柱、ツメ、ノンブル
- デザイナーの魅せドコロとなることも



『Linuxサーバーセキュリティ徹底入門』
(2013, 翔泳社)

AHF課題3: 版面外の要素調整

- AH Formatter/CSSの紙面要素概念



AHF課題3: 版面外の要素調整

- 紙書籍 固有の話
- 版面の中を操作するCSSは豊富
- しかし、版面外の指示があまり細かく書けない
- 「左上」などの既定位置をまたがりにくい
- content属性やstring関数の融通が効かない
- 改善予定？

まとめ(1)

- InDesignでは手間のかかることが、AH Formatterなら華麗にこなせるケースがある
  - 完全な自動化
  - フロート、サイドバイサイド、バーティカルリズム、etc...

まとめ(2)

- AH Formatter/CSSで出版社水準を満たす技術書組版には、まだ解決の必要な課題あり
  - 組設定、見開き、版面外描画
  - これまでのアンテナハウスの活発な開発実績を見るに、解決は遠くないと期待

まとめ(3)

- 書籍のマスターデータには、扱いやすい原稿フォーマットの導入を
  - 原稿からの一方向変換での組版/EPUB作成手法
  - HTMLを中心にするのは疑問
  - Re:VIEW形式オススメ
    https://github.com/kmuto/review

まとめ

- InDesignでは手間のかかることが、AH Formatterなら華麗にこなせるケースがある
- AH Formatter/CSSで出版社水準を満たす技術書組版には、まだ解決の必要な課題あり
- 書籍のマスターデータには、扱いやすい原稿フォーマットの導入を

補足

- 株式会社トップスタジオ
  - http://www.topstudio.co.jp/
- Re:VIEW
  - https://github.com/kmuto/review
- 『はじめてのReVIEW』（TechBooster）
  - https://techbooster.booth.pm/items/12746