

# Moby-Dick 『白鯨』 を組版する

---

はじめに	2
段階的に近似	2
ページ画像からのスタイル付け	3
前付	8
タイトルページ	8
本のタイトル	8
目次	10
「語源」と「抜粋」	12
本文	13
章区切り	14
脚注	16
重複した脚注	17
脚注サイズ	19
ブロック	20
ウィドウとオーファン	20
ページの最後にハイフン	22
テキスト	22
イタリック体とスモールキャピタル	22
カールした引用符	23
連続した em ダッシュ	25
行グリッド	26
ヘッダとフッタ	28
おわりに	29
参考文献	30

## はじめに

この文書では、ハーマン・メルヴィルによる Moby-Dick 『白鯨』 アメリカ初版をフォーマットするために開発されたスタイルシートの側面について説明します。これらのスタイルシートは、XSL-FO 用のスタイルシートを開発するアプローチのうちのひとつの方法を示しており、また、いくつか Antenna House Formatter 拡張機能の使用方法も紹介しています。

Moby-Dick 『白鯨』 (1)のソースは、Wright American Fiction プロジェクト (3) の TEI エンコード XML (2) です。

スタイルシートは、Antenna House Formatter V7.0 (5) の自動分析機能 (4) の使用方法を示すプロジェクト用に開発されました。そのため、最初は本文ページのテキストブロックを正しくすることに重点が置かれました。テキストブロック以外のすべて(ヘッダとフッタ、前付、後付に広告)のスタイルは、当初、初版で使用された書式設定の大まかな近似として開発されました。時間が経つにつれて、印刷された初版をより正確に模倣するためにスタイルが洗練されました。

## 段階的に近似

XSL-FO または CSS で書式設定するためのスタイルシートを開発することは、通常、最終結果に向けて段階的な近似を開発するプロセスです。これは、文書の外観がその場で開発されている場合でも、設計概要に従って開発されている場合でも、Moby-Dick 『白鯨』のように既存の文書に一致するように開発されている場合でも当てはまりません。

スタイルシートの最初のドラフトでは、最終結果の大まかな近似値しか生成されない可能性があります。臨機応変に開発している場合は、とにかくその時点では最終的な見た目については決まっていません。デザイン概要に従って開発している場合、最初の組版バージョンには正しいページ サイズと正しいフォント、および主要なタイトルと段落のフォントサイズが含まれる可能性があります。目次などのコンテキスト固有のスタイルが省略されるかもしれません。コンテンツ、インデックス、表、入れ子になったリストなど。既存のドキュメントに一致するスタイルを開発する場合も同様です。

通常、その後の一連の変更を加えてレビューし、スタイルを最終結果に近づけます。もちろん、これは、臨機応変に開発している場合にも当てはまります。望んでいる結果が得られたと言うまで、最終結果は分からないからです。これは、設計概要に従って開発する場合と、既存のドキュメントに一致するように開発する場合の両方に当てはまります。これは、まだ処理していないことがわかっている追加のコンテキストがあり、おそらく、あなたも設計者も予期していなかった追加のコンテキストが存在するためです。これらには、さまざまなタイプの入れ子になったリスト、タイトルの直後の図や表の処理、Moby-Dick 『白鯨』の場合、物語のキャラクターである Queequeg のマークや舞台演出や曲の処理などが含まれる場合があります。

もちろん、変更を加え続けることで最終結果に近づくはずですが。実際には、一部の変更はやり直す必要があり、一部の変更では新たな問題が発生することもあります。全体的な動きは最終結果に近づきつつあります。

### ページ画像からのスタイル付け

特に Moby-Dick 『白鯨』の前付の初期スタイルは、初版のページの写真を対応するページの背景画像として設定し、それに合わせて XSL-FO を調整することで洗練されました。次の画像は、Antenna House Formatter GUI で開いたときの、初版のタイトルページの写真をページ背景として組版したタイトルページを示しています。

## Formatting Moby-Dick



ページ画像を背景として使用する一連の手順は次のとおりです。

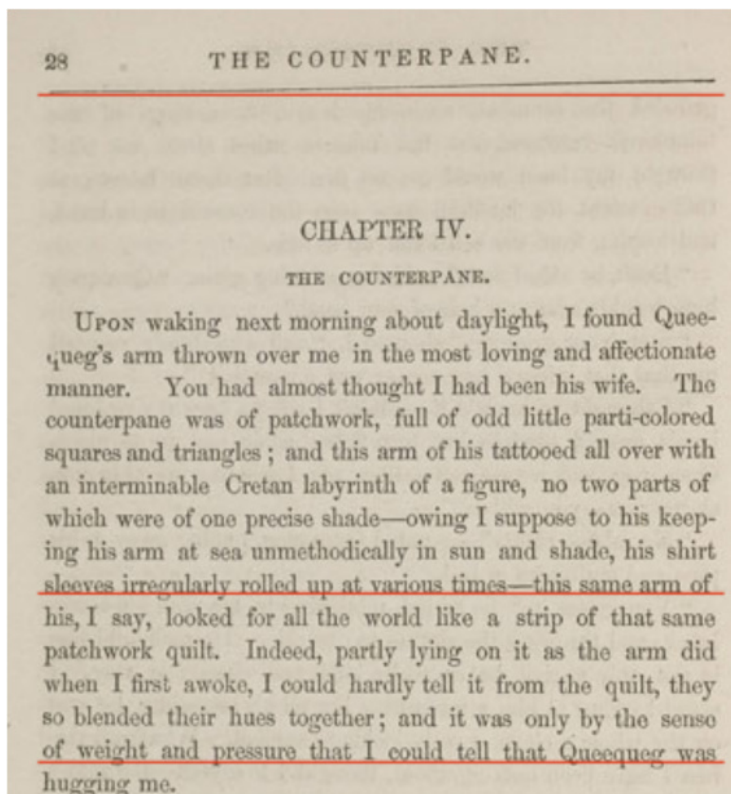
- 1.

XSL-FO を変更して、背景画像を持つ各 `fo:simple-page-master` に `axf:bleed` プロパティと `axf:crop-offset` プロパティを追加します。例えば次のように指定します。

```
<fo:simple-page-master master-name="First-PageMaster"
    page-height="7.375in"
    page-width="4.78in"
    axf:bleed="0.5in"
    axf:crop-offset="0.5in">
```

2. 必要に応じて、テキストができるだけ水平になるようにページ画像を回転します。

初版は今から 170 年前のもので、利用可能なページ画像は個々のページをスキャンしたものではなく、製本された本のページの写真です。その結果、ページの状態またはページを撮影したときの紙の湾曲により、スキャンしたテキストは必ずしも完全に平行になるとは限りません。次の図は、変動が発生する可能性があることを示しています。赤い線は平行ですが、テキストは平行ではありません。



3. fo:simple-page-master のいずれかの背景画像として、組版したページに一致するように拡大縮小したページ画像を指定します。

```
<fo:simple-page-master
    master-name="First-PageMaster"
    page-height="7.375in"
    page-width="4.875in"
    background-image="page-images/MD_Amer_0038.jpg">
```

```
axf:background-size="5.21in"  
background-position="-0.12in -0.15in"  
axf:bleed="0.5in"  
axf:crop-offset="0.5in">
```

または、ページを生成する **fo:page-sequence** 上で次のようにします。

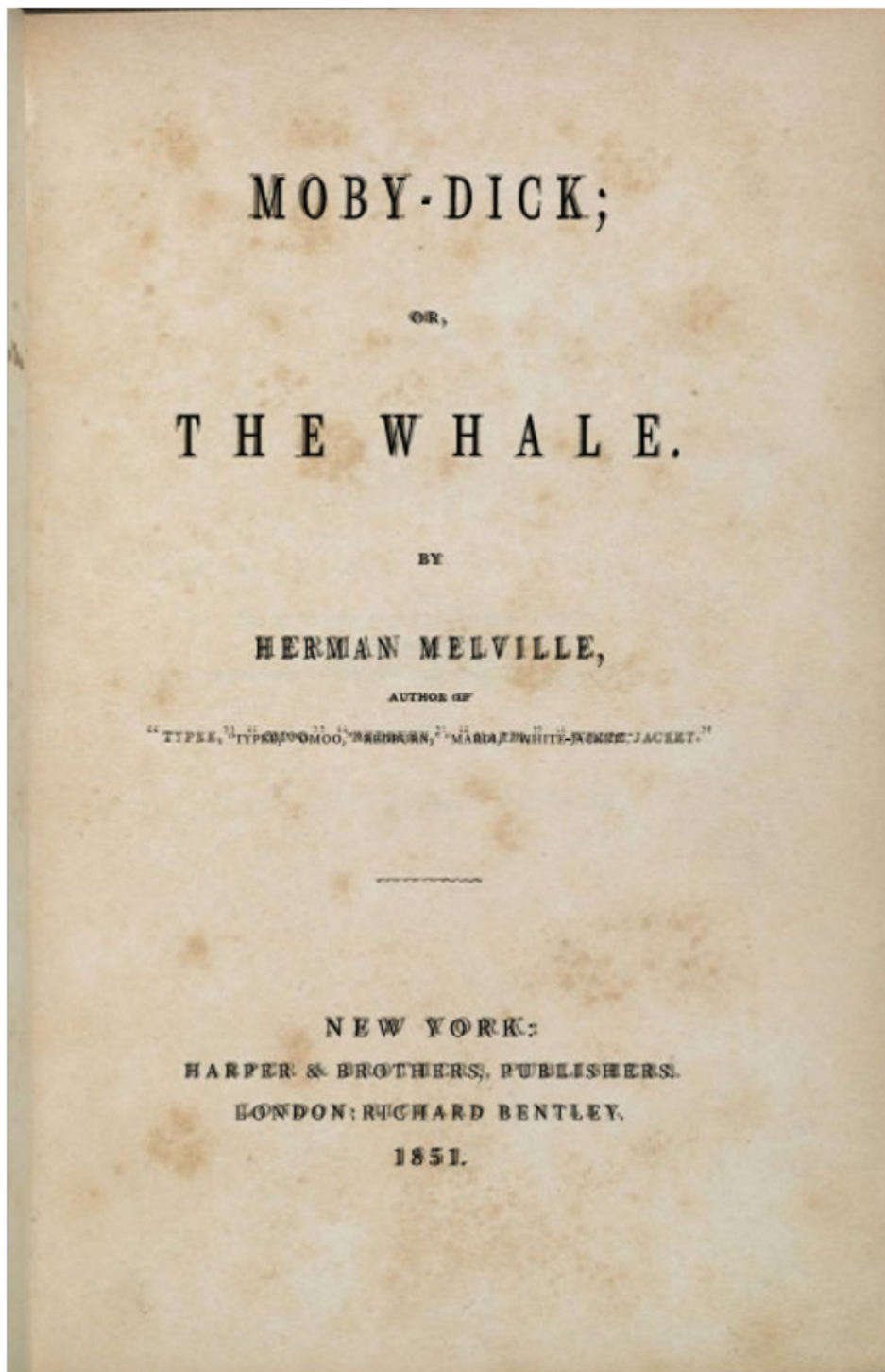
```
<fo:page-sequence  
  master-reference="CoverFrontMaster"  
  background-image="page-images/MD_Amer_0019.jpg"  
  axf:background-size="5.7in"  
  background-position="-0.7in -0.3in">
```

初版のページ画像は写真であるため、各画像内のページのサイズや位置にはかなりのばらつきがありました。正しいサイズと位置を取得するには、**XSL-FO** を変更して **Antenna House Formatter GUI** で結果を表示し、満足のいく結果が得られるまでこのプロセスを繰り返すという反復的なプロセスが必要でした。

**Antenna House Formatter GUI** で「ボーダー表示」を有効にすると、背景画像の調整方法を判断しやすくなります。

4. 組版したドキュメントが初版のページと十分に一致するまで、**XSL-FO** を繰り返し変更し、**Antenna House Formatter GUI** で表示します。
5. **XSL-FO** を生成するためのスタイルシートを変更して、手動で、取得した **FO** とプロパティを再作成します。

結果は、元のものに非常に近いものになると思います。



初版の前付のさまざまな部分では、フォント、フォントサイズ、文字と単語の間隔がかなり異なっています。これに、不正確な背景画像のサイズと位置が加わり、同じプロパティの値の範囲が異なるページの異なる場所に適用されることになりました。時間が許せば、これらを合理化して、より少なく、より一貫した値を使用しながら、初版のページを十分な精度で再現できるはずですが、結局のところ、初版は固定のフォントセットと文字間に追加できる空白の固定増分で印刷されました。1851年のアメリカ

ではフォントサイズなどがポイント単位で指定されていたとは考えられませんが、サイズは内部的には一貫していたと考えられます。

### 前付

Moby-Dick『白鯨』の前付は次のとおりです。

- タイトルページ
- 著作権
- 献辞
- 目次
- フライタイトル
- 語源
- 抜粋

### タイトルページ

前に示したように、タイトルページをかなり正確に再現することができます。

#### 本のタイトル

本のタイトルのマークアップには、組版したタイトルを正確に再現するのに十分な情報が含まれていません。

```
<docTitle>
  <titlePart>MOBY-DICK;</titlePart>
  <titlePart type="sub">OR, THE WHALE.</titlePart>
</docTitle>
```

さらに、本のタイトルはフライタイトルページ上で等しく組版されますが、そのマークアップは組版に対応していません。

```
<div type="fly_title">
  <head>MOBY-DICK; OR, THE WHALE.</head>
</div>
```

スタイルシートは Moby-Dick『白鯨』に固有であるため、マークアップを無視して `xsl:analyze-string` を使用し、タイトルテキストの一部の FO を生成する方が簡単でした

```
<xsl:template match="docTitle | div[@type = 'fly_title']/head"
  priority="5">
  <fo:block
    font-size="24pt"
    letter-spacing="0.37em"
    line-height="1"
    text-align="center">
```



```

        font-stretch="extra-condensed">
<xsl:analyze-string
  select="normalize-space(.)"
  regex="OR, ">
<xsl:matching-substring>
  <fo:block
    font-size="8pt" font-variant="all-small-caps"
    font-stretch="normal"
    letter-spacing="0.125em" space-before="30pt">
    <xsl:value-of select="." />
  </fo:block>
</xsl:matching-substring>
<xsl:non-matching-substring>
  <fo:block axf:letter-spacing-side="start">
    <xsl:if test="contains(., 'THE WHALE.')">
      <xsl:attribute name="space-before" select="'30pt'"
/>
      <xsl:attribute name="letter-spacing"
select="'0.9em'" />
    </xsl:if>
    <xsl:analyze-string
      select="."
      regex="\.| ">
    <xsl:matching-substring>
      <fo:inline letter-spacing="0.3em">
        <xsl:value-of select="." />
      </fo:inline>
    </xsl:matching-substring>
    <xsl:non-matching-substring>
      <xsl:value-of select="." />
    </xsl:non-matching-substring>
  </xsl:analyze-string>
</fo:block>
</xsl:non-matching-substring>
</xsl:analyze-string>
</fo:block>

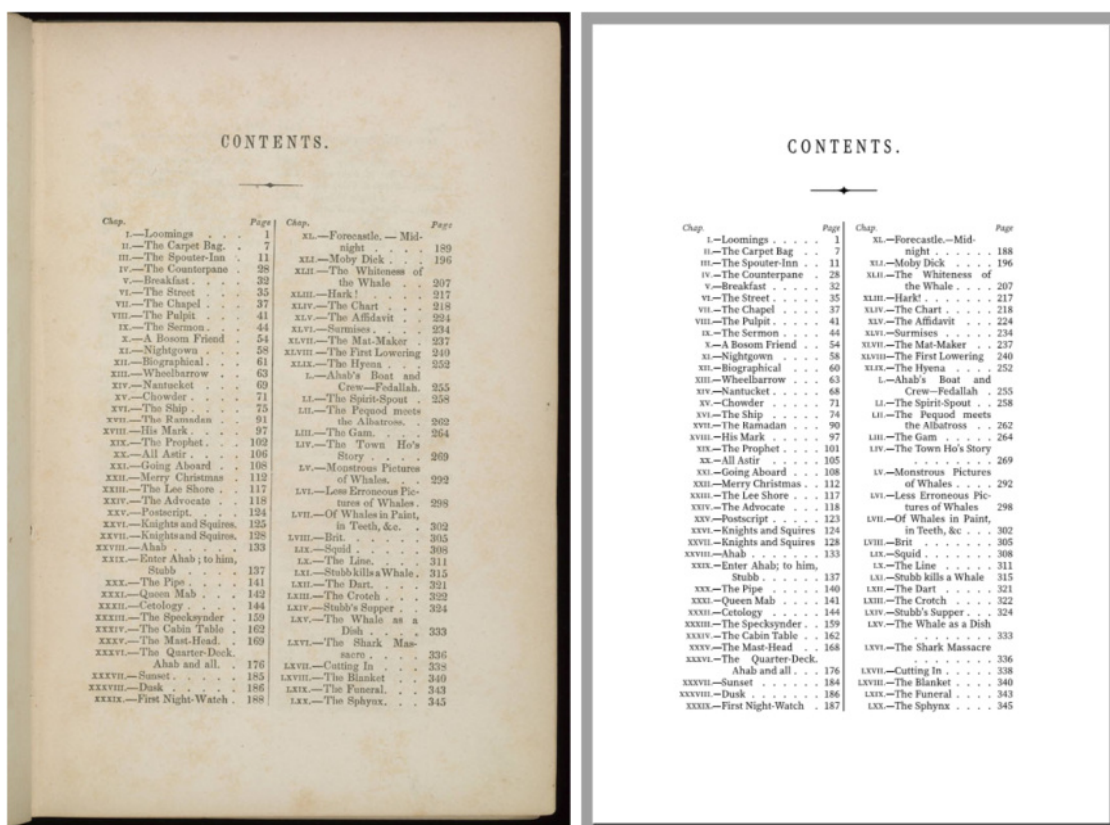
```

このドキュメントは、いくつかの小さな例外を除いて、完全に Source Serif Pro で組版されています。このフォントはオープンソースであり、初版の段落に使用されたフォントとかなり一致しています。ただし、初版では、タイトルページ（および他の一部のタイトル）では、フォントの細字とスモールキャピタルの両方が使用されています。Source Serif Pro には細字バージョンがありませんので、したがって、細字は、font-stretch プロパティ（たとえば、font-stretch="extra-condensed"）を設定し、Antenna Formatter に依存して各文字の幅を調整することによって実現されます。Source Serif Pro には実際のスモールキャピタル（6）がありますが、XSL 1.1 で定

義されている「font-variant="small-caps"」では、小文字にのみスモールキャピタルが使用されます。タイトルのスモールキャピタルは XML では大文字として表されるため、Antenna house Formatter の拡張機能 font-variant="all-small-caps"を使用して、ソース内の大文字をスモールキャピタルとして組版する必要があります。

初版のタイトルの多くは、文字間隔のある文字を使用しています。文字間隔は letter-spacing プロパティで指定します。letter-spacing="0.37em" などの値は、ページ画像を背景として使用したときの外観に合わせて試行錯誤して導き出されました。ただし、初版では、アルファベットとそれに続く句読点の間の文字間隔が、ふたつのアルファベット間の文字間隔よりも狭い場合があります。解決策は、xsl:analyze-string を使用して、それらの文字の周囲に異なる letter-spacing の値を持つ fo:inline を生成することです。スタイルシートのその他の改良点は、Antenna House Formatter の拡張機能 axf:letter-spacing-side も使用して、追加されるすべての空白が開始側に追加されるアルファベット文字と文字の間にあるようにすることです。したがって、アルファベット文字とそれに続く句読点文字の間の空白には影響しません。

目次



目次は 2 段組みで構成されています。リストとしてマークアップされていますが、初版の書式を再現できるように 4 段組みの表として表示されます。

目次の TEI は次のように始まります。

```
<div type="contents">
  <pb n="v (Table of Contents) " xml:id="VAC7237-00000003"/>
  <head>CONTENTS.</head>
  <list>
    <item>I.-Loomings. <ref target="VAC7237-00000013"
rend="right">1</ref>
    </item>
    <item>II.-The Carpet Bag. <ref target="VAC7237-00000017"
rend="right">7</ref>
    </item>
```

<ref> 要素の内容は、初版の各章のページ番号です。ただし、target 属性は、各見開き 2 ページの開始を示す <pb> milestone 要素を指します。相互参照の少なくとも 1 つが、その章の最初のページの後の見開きを指していることが判明し、修正する必要がありました。さらに多くの修正点が見つかるかもしれません。

XSL-FO 版では初版の改ページを再作成しようとしていないため、相互参照はいずれにせよ使用できませんでした。目次エントリから生成された PDF 内の章への相互参照は、目次リスト内の各章のリスト項目の位置から決定されます。

```
<!-- Every chapter has a generated ID, and 'EPILOGUE.' is the
only
  ToC entry without a page number. -->
<xsl:variable
  name="target"
  select="if (exists(ref))
    then concat('chapter-', position())
    else 'epilogue'"
  as="xs:string" />
```

各章のマークアップは「<div type="chapter">」で始まりますが、一部の <pb> milestone 要素が章の間に表示されるため、各章の ID の生成に「position()」を使用できませんでした。

```
<fo:block
  id="{@type}-{count(preceding::div[@type = current()/@type])
+ 1}">
```

目次エントリのさまざまな部分を揃えるために、目次は 4 段組みの表として組版されています。

- 章番号 (スモールキャピタル処理のローマ数字)
- Em ダッシュ
- 章のタイトルとリーダドット

XSL-FO を使用すると、リーダドットの位置合わせと間隔調整が簡単です。

```
<fo:leader leader-pattern="dots"
           leader-pattern-width="1em"
           leader-alignment="end" />
```

- ページ番号

## 「語源」と「抜粋」

「語源」と「抜粋」の各セグメントは、それぞれ、引用文が続く導入の物語ページで構成され、「語源」では表が続きます。初版ではタイトルに使用されているフォントが統一されていないため、それぞれ別のテンプレートが必要でした。

ETYMOLOGY.

ETYMOLOGY

EXTRACTS.

EXTRACTS.

「語源」と「抜粋」の両方において、各引用には出展（著作権者の名前）があり、各出展（著作権者の名前）は引用資料に続くようにマークアップされています。

```
<cit>
  <q>
    <p>"Very like a whale."</p>
  </q>
  <bibl>
    <title>Hamlet</title>.</bibl>
</cit>
```

ただし、最後の行に十分な空白がある場合、出展（著作権者の名前）は引用と同じ `<fo:block>` で組版されます。

“Very like a whale.”

*Hamlet.*

最後の行または出展（著作権者の名前）が長すぎるために十分な空白がない場合、出展（著作権者の名前）は次の行に組版されます。

同じブロック内に出版（著作権者の名前）を配置することは、デフォルト処理の一部として `<bibl>` を組版せず、代わりに `<q>` の処理時にそのコンテンツを明示的に選択するという一般的な XSLT パターンによって処理します。

```
<xsl:template match="q/p">
  <fo:block>
    <xsl:apply-templates />
    <xsl:if test="position() = last() and
      exists(..following-
sibling::*[1][self::bibl])">
      <fo:leader leader-pattern="space"/>
      <fo:leader leader-pattern="space"
        leader-length.optimum="100%"/>
      <fo:inline-container padding-left="2em"
        padding-right="0.125in"
        max-width="80%" text-indent="0">
        <fo:block text-align="right">
          <xsl:apply-templates
            select="..following-sibling::*[1]/node()" />
        </fo:block>
      </fo:inline-container>
    </xsl:if>
  </fo:block>
</xsl:template>

<xsl:template
  match="bibl[exists(preceding-sibling::*[1][self::q[p]])]"
  priority="5" />
```

引用の最後の行または次の行に出展（著作権者の名前）を配置する場合は、ふたつの `<fo:leader>` を使用する一般的な XSLT パターンによって処理します。

## 本文

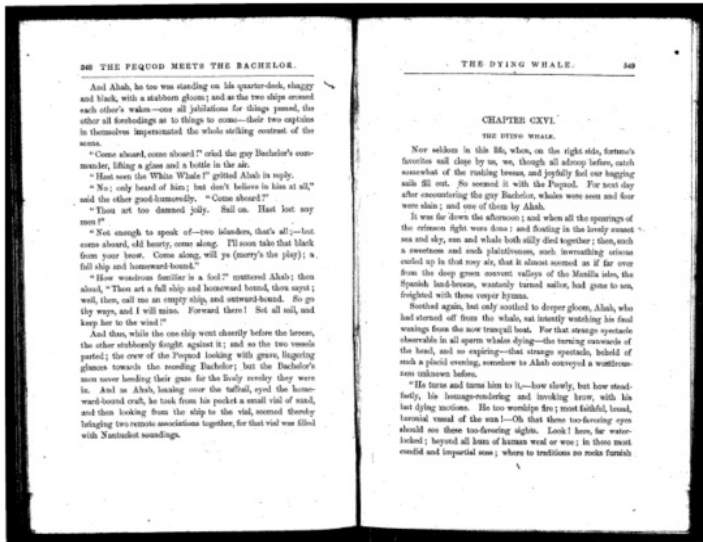
Moby-Dick 『白鯨』の大部分は 135 章からなり、そのほとんどがテキストで構成されています。メルヴィルを研究する学者は章の構造にパターンを見つけることを好みます (7)が、Moby-Dick 『白鯨』を組版する際に最も役立つのは、段落のようなテキストのブロックとその他のコンテンツとの区別です。

段落以外のコンテンツには次のものが含まれます。

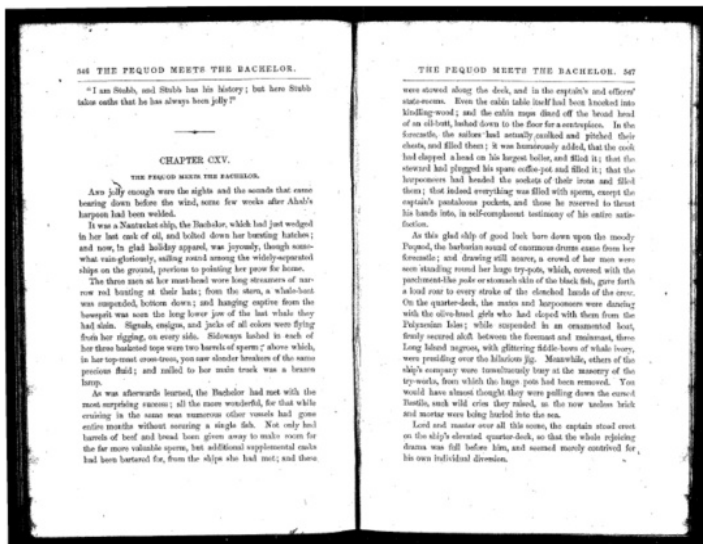
- A single graphic (for Queequeg's mark) 単一のグラフィック (登場人物 Queequeg のマーク用)
- 墓石の碑文
- 歌と詩
- まるで演劇のようなスピーチと舞台演出

### 章区切り

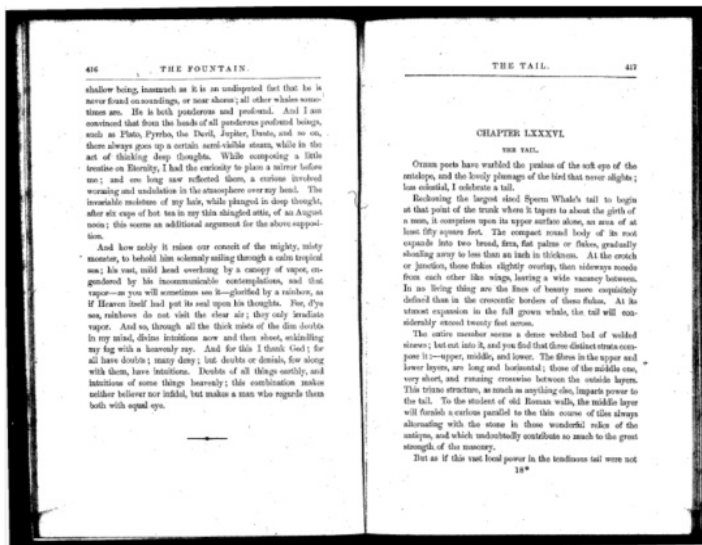
初版で章がページの下部近くで終わる場合、次の章は章のタイトルの前に空白を設けて次のページから始まります。



章がページの下部近くで終わらない場合は、章のタイトルの前に追加の章区切りが印刷されます。問題を複雑にしているのは、章区切りと章のタイトルの間の空白が、新しいページで始まる章の前の空白よりも小さいことです。



章の終わりにページの下部にいくらかの空白が残っていても、章区切りと章のタイトルのための空白が足りない場合、章区切りは章の最後に印刷され、次の章は次のページから始まります。



初版が手動で作成されたときは、必要なときに必要な場所に章区切りを追加するのは簡単でした。しかし、XSL-FOを使用した完全自動の組版では、それほど簡単ではありません。ドキュメントを組版する前にページ区切りがわからないため、必要な数の章区切りを挿入することはできません。また、XSL 1.1 勧告では、ページ上の領域の位置に基づいた条件付き処理はサポートされていません

Antenna House Formatter を使って、次の ふたつのことが可能になります。まず、`axf:suppress-if-first-on-page` 拡張プロパティにより、Antenna House Formatter はページ上部の章タイトルの章区切りを非表示にします。次に、章区切りの `<fo:block>` の標準機能の `space-after.precedence="force"` により、章区切りが存在する場合に章区切りと章タイトルの間の距離が適切に保たれ、章区切りが非表示になっているか前のページにある場合は、章タイトルの異なる `space-before` の値を適用できるようになります。

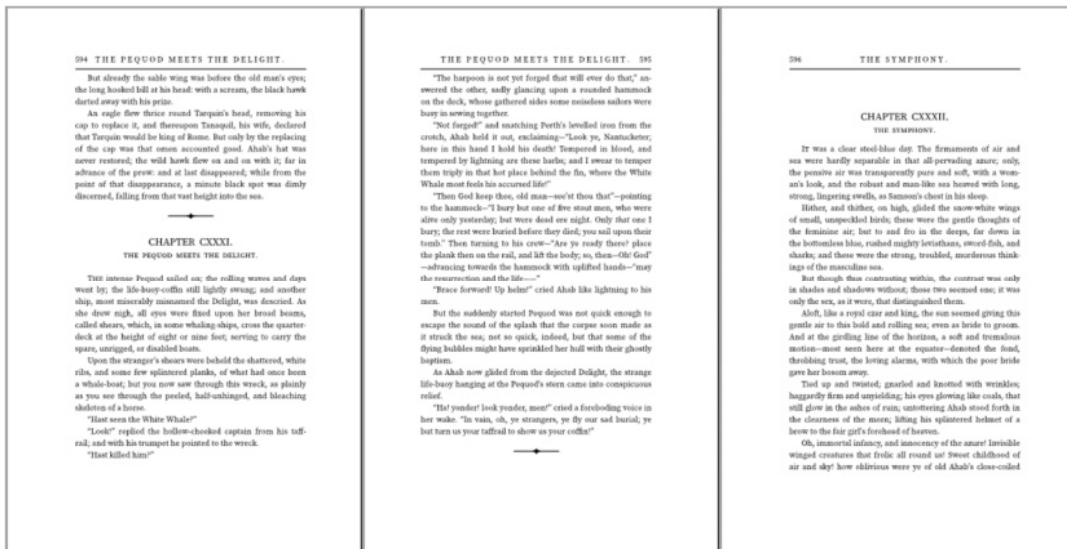
```
<xsl:template
  match="div[@type = 'chapter'][exists(head[@type = 'sub'] |
fw[@type = 'head'])]">
  <xsl:if test="exists(preceding-sibling::div[@type =
current()/@type])">
    <fo:block axf:suppress-if-first-on-page="true" text-
align="center"
      padding-top="0.125in"
      space-after="0.2in" space-after.precedence="force"
      axf:baseline-grid="none"
      axf:baseline-block-snap="none">
      <fo:external-graphic src="images/separator.svg" />
    </fo:block>
  </xsl:if>
  <fo:block
```

## Formatting Moby-Dick

```

    id="{@type}-{count(preceding::div[@type =
current()/@type]) + 1}">
    <fo:marker marker-class-name="Chapter-Title">
      <xsl:apply-templates
        select="(fw[@type = 'head'], head[@type =
'sub'])[1]/node()"
        mode="marker" />
    </fo:marker>
    <fo:block-container
      axf:baseline-grid="none"
      axf:baseline-block-snap="none"
      keep-together.within-page="always"
      keep-with-next.within-page="always"
      space-before="{if (exists(preceding::div[1][@type =
'chapter']))
        then '0.5in'
        else '0.72in'}"
      space-before.conditionality="retain">
      <xsl:apply-templates select="head" />
    </fo:block-container>
    <xsl:apply-templates select="*" except head" />
  </fo:block>
</xsl:template>

```



## 脚注

脚注は、本文の枠外の <note> を参照する脚注マーカと脚注コンテンツを含む <ref> としてマークアップされます。

<p>

<emph>Whaling not respectable?</emph> Whaling is imperial! By



old English

```
statutory law, the whale is declared "a royal fish."<ref
rend="super"
target="#note_001" xml:id="return_001">*</ref>
<note place="foot" xml:id="note_001">
  <p><ref target="#return_001">*</ref>See subsequent chapters
for something more on this head.</p>
</note>
</p>
```

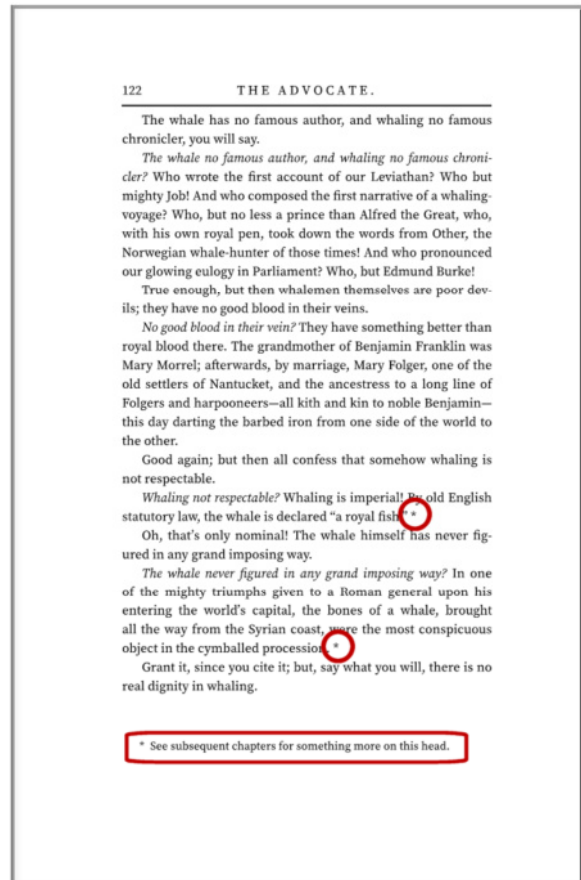
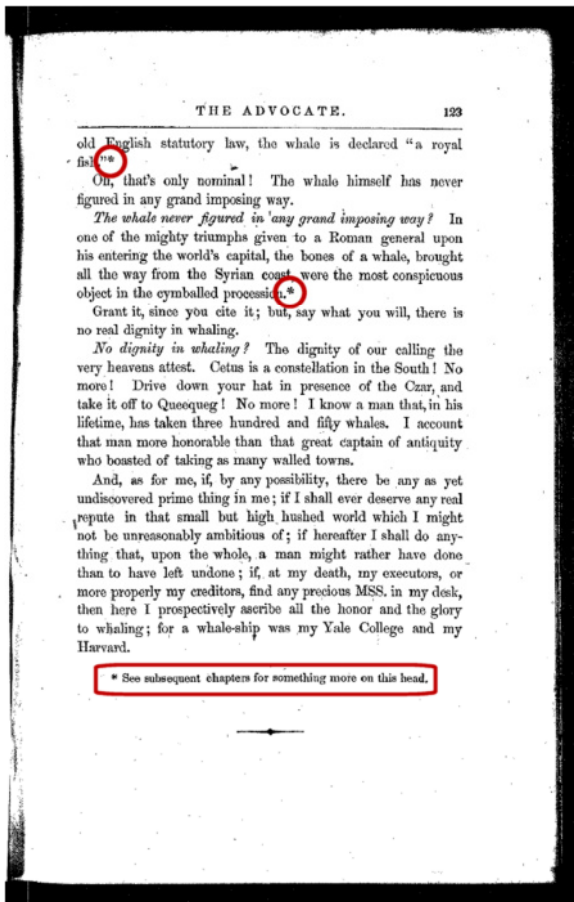
XSL-FO <fo:footnote> には、脚注マーカ用の <fo:inline> と脚注コンテンツ用の <fo:footnote-body> の両方が含まれているため、XSLT スタイルシートはドキュメント内で <note> が出現する場所ではそれを処理せず、代わりに各 <ref> によって参照される注釈を ‘key()’ を使用して検索することで、<note> のコンテンツを組版します。

```
<xsl:template match="note[@place = 'foot']" />

<xsl:template match="ref[exists(key('footnote',
                                     substring-after(@target,
                                     '#')))]"
               priority="5">
  <fo:footnote
    id="{@xml:id}"
    axf:suppress-duplicate-footnote="true">
    <fo:inline>
      <fo:basic-link
        internal-destination="{substring-after(@target,
        '#')}">
        <xsl:value-of select="." />
      </fo:basic-link>
    </fo:inline>
    <fo:footnote-body
      id="{substring-after(@target, '#')}"
      font-size="7pt"
      line-height="10pt">
      <xsl:apply-templates
        select="key('footnote',
                   substring-after(@target, '#'))/node()" />
    </fo:footnote-body>
  </fo:footnote>
</xsl:template>
```

### 重複した脚注

初版の1 ページ内には、同じ脚注への参照が ふたつあります。



TEI XML は脚注テキストを繰り返します。

```

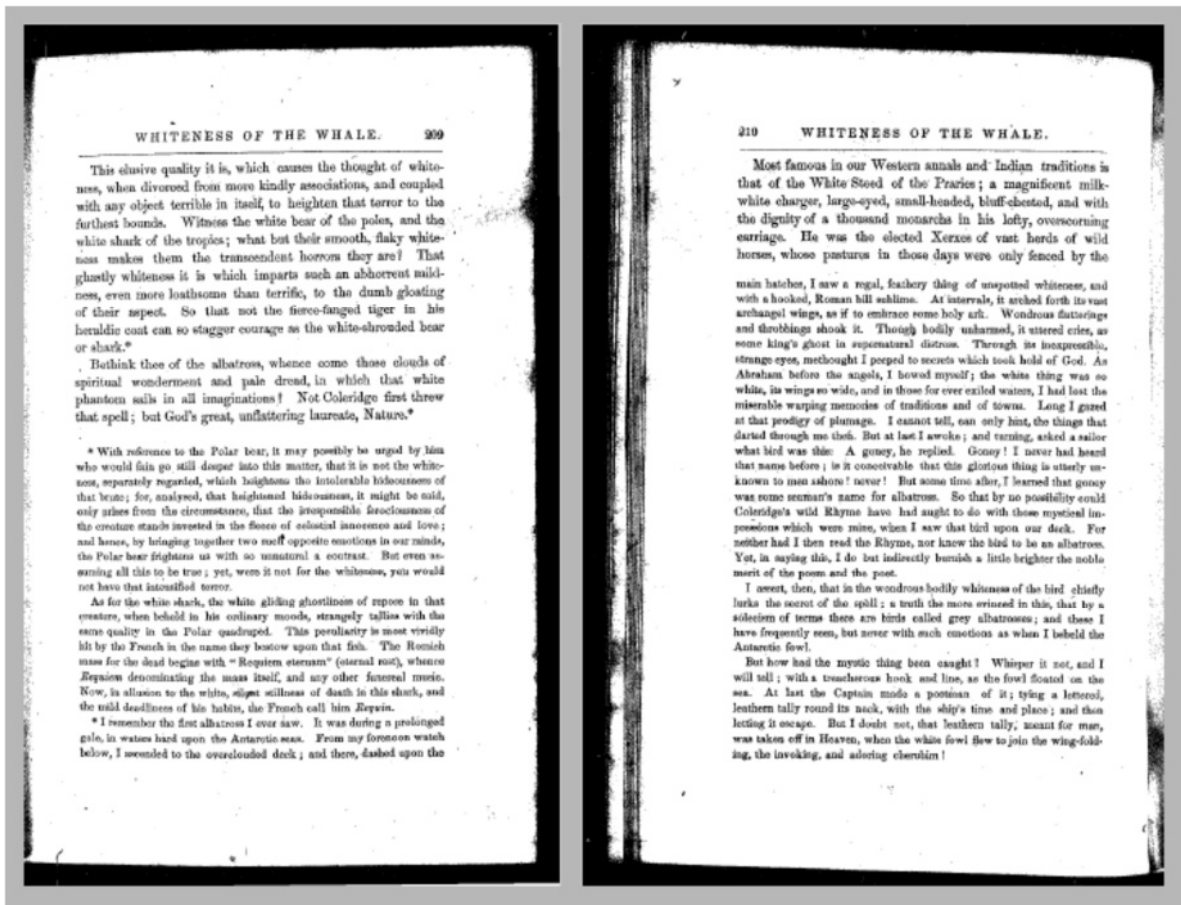
<p>
  <emph>Whaling not respectable?</emph> Whaling is imperial! By
  old English
  statutory law, the whale is declared "a royal fish."<ref
    rend="super" target="#note_001" xml:id="return_001">*</ref>
  <note place="foot" xml:id="note_001">
    <p><ref target="#return_001">*</ref>See subsequent chapters
    for something more on this head.</p>
  </note>
</p>
...
<p>
  <emph>The whale never figured in any grand imposing
  way?</emph> ... cymballed procession. <ref rend="super"
    target="#note_002" xml:id="return_002">*</ref>
</p>
<note place="foot" xml:id="note_002">
  <p><ref target="#return_002">*</ref>See subsequent chapters
  
```

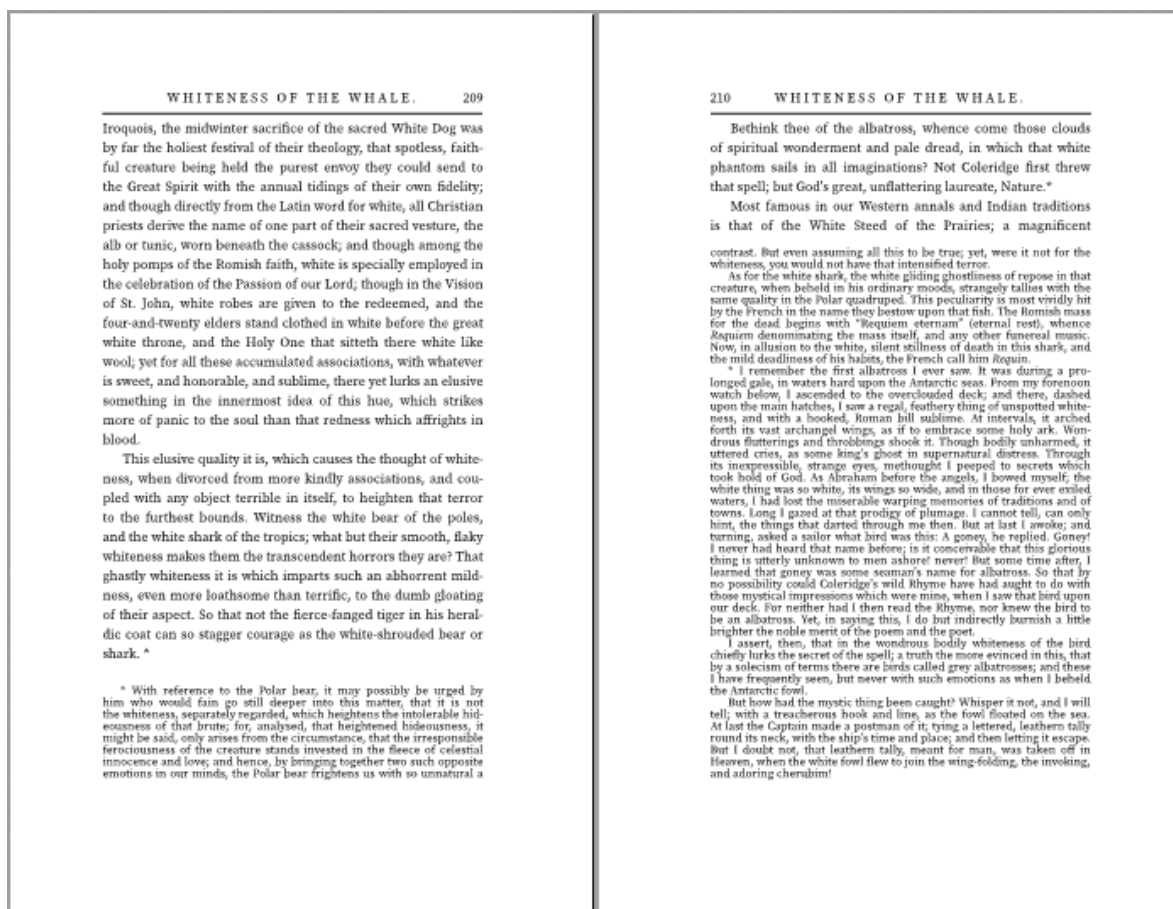
for something more on this head.</p>  
</note>

厳密な XSL 1.1 実装では両方の脚注がレンダリングされますが、`axf:suppress-duplicate-footnote` 拡張プロパティにより、両方の脚注が同じページにある場合、Antenna House Formatter は脚注をひとつだけ生成します。

### 脚注サイズ

Moby-Dick 『白鯨』にはクジラサイズの大きな脚注もいくつか含まれています。





実行すべきだったかもしれないいくつかの作業は実は不要でした。

- 初版では、これらふたつの脚注は同じページから始まり、2番目の脚注が2ページ目に続きます。それでも、初版では両方の脚注に同じ「\*」脚注マークが付いています。

第1版のマークはすべて同じであるため、脚注マークのシーケンスを生成し `axf:footnote-number` および `axf:footnote-number-citation` 拡張要素を使用する必要はありません。

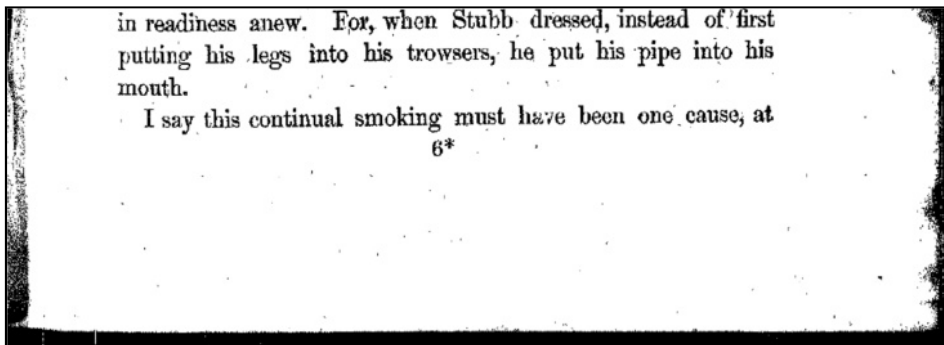
`axf:footnote-max-height` を使用して脚注の高さを制限することは可能ですが、組版された脚注の高さは初版の高さと同程度であるため、これも必要ありませんでした。

## ブロック

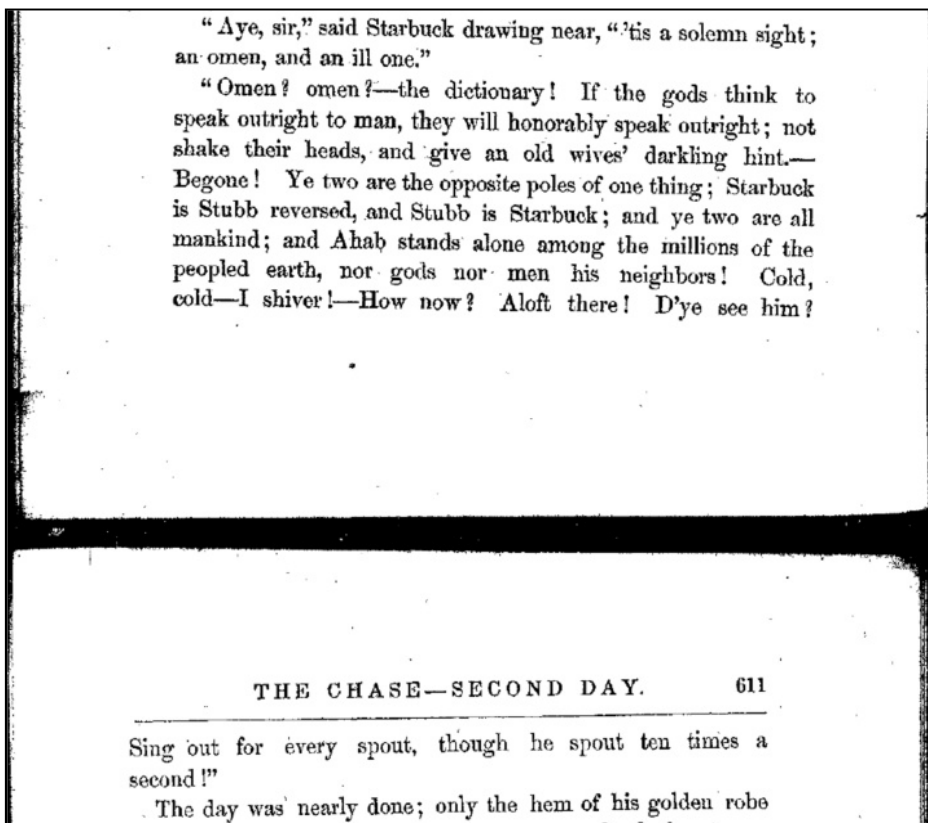
### ウィドウとオーファン

オーファンとは、改ページ前の行数が少なすぎることであり、ウィドウとは、改ページの後の行数が少なすぎることです。

初版には、単一行のオーファンが複数あります。



しかし、ページ上部の 1 行は 1 行のセリフのみです。意図的に 2 行のウイドウにした行がいくつあるのかはわかりません。たとえば、610 ページは広い行間を空けたテキストで終わり、611 ページは段落の最後の 2 行で始まります。



行間隔が広いのは、初版の「?」と「!」の前の空白によるものですが、初版に比べて、Antenna House Formatter で段落を組版したときは行数が少なくなりました。

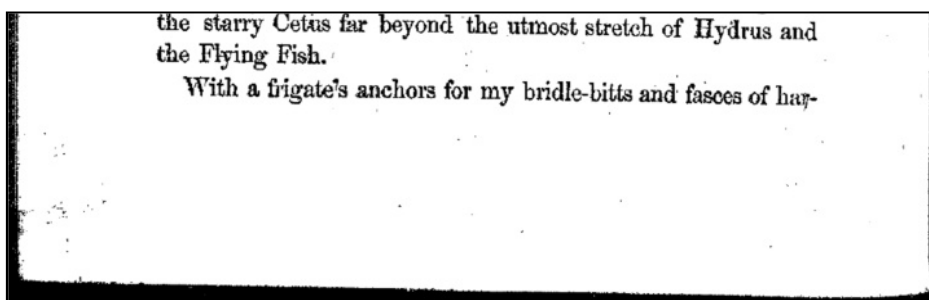
“Omen? omen?—the dictionary! If the gods think to speak outright to man, they will honorably speak outright; not shake their heads, and give an old wives’ darkling hint.—Begone! Ye two are the opposite poles of one thing; Starbuck is Stubb reversed, and Stubb is Starbuck; and ye two are all mankind; and Ahab stands alone among the millions of the peopled earth, nor gods nor men his neighbors! Cold, cold—I shiver!—How now? Aloft there! D’ye see him? Sing out for every spout, though he spout ten times a second!”

同様に、初版の 371 ～ 373 ページの段落は 26 行ですが、Antenna House Formatter で 1 ページに組版すると 25 行になります。最初の 4 行は初版と同じですが、その後は異なります。

組版バージョンでは、XSL-FO 1.1 のデフォルトである `orphans="2"` および `widows="2"` を使用します。

## ページの最後にハイフン

初版にはハイフンで終わるページが複数あります。



組版バージョンでは、`fo:root` に `hyphenation-keep="page"` が指定されているため、単語は改ページをまたいでハイフンで区切られません。オプション設定ファイルの `hyphenation-keep-mode` の設定は上書きされないため、Antenna House Formatter は最後の行全体ではなく、ハイフンで区切られた単語のみを次のページに押し出します。

## テキスト

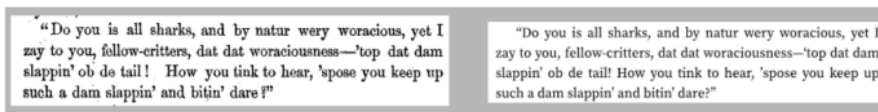
### イタリック体とスモールキャピタル

イタリック体とスモールキャピタルのテキストのマークアップは、組版を適切に行うために修正する必要がありました。元の TEI XML では、イタリック体のテキストは、先頭に空の `<hi rend="i"/>` 要素でマークされていましたが、イタリック体のテキスト終了位置は示されていませんでした。イタリック体のテキストを囲まない方がテキスト分析が容易になるという意見もあるかもしれませんが、外国語 (または外国語と思われる単語) は、開始タグと終了タグでマークアップされていました (例: `<foreign xml:lang="LAT">Folio</foreign>`)。

初版のスモールキャピタルのテキストは、追加のマークアップなしで大文字として TEI XML に含まれていました。スモールキャピタルにするべきテキストを見つけてマークアップを追加し、テキストを大文字と小文字の混合に変更する必要がありました。たとえば、「THE」は「<hi rend="small-caps">The</hi>」になります。

## カールした引用符

Moby-Dick『白鯨』では、一重引用符と二重引用符の両方が頻繁に使用されています。これには、英語を母国語としない人の話し方に合わせて、ブロークンな英語の文字をアポストロフィで置き換えることも含まれます。初版では、左と右の引用符が明らかに異なっています。



ただし、TEI ソース XML では、すべての引用符は直線の形状です。

```
<p>"Do you is all sharks, and by natur wery voracious, yet I zay to you, fellow-critters, dat dat voraciousness—'top dat dam slappin' ob de tail! How you tink to hear, 'spose you keep up such a dam slappin' and bitin' dare?"</p>
```

直線の引用符をカールした引用符に変換することは、最初は簡単に思いましたが、強調されたテキストの前の引用符と、引用符テキストの先頭の左一重引用符と、省略された文字を示す単語の先頭の右一重引用符の違いによって複雑になりました。

```
<!-- Convert single and double quotes to 'curly' quotes. -->  
<xsl:template match="text()" name="ahf:text">  
  <xsl:param name="text" select="." as="text()" />  
  
  <xsl:value-of select="ahf:text($text)" />  
</xsl:template>  
  
<xsl:function name="ahf:text" as="xs:string">  
  <xsl:param name="text" as="text()" />  
  
  <!-- The replacement that depends on the current node must be  
  first. -->  
  <xsl:variable  
    name="text"  
    select="if (matches($text, '".$') and  
              empty($text/following-sibling::node()))  
            then replace($text, '".$', '&rdquo;')  
            else $text"
```

## Formatting Moby-Dick

```
        as="xs:string" />
    <!-- Moby-Dick uses broken English for speech from non-native
        speakers of English. The speech can include words with
the
        dropped initial vowel indicated by a right single-quote.
        Handle those before replacing any &apos; with left
        single-quotes. -->
    <xsl:variable
        name="text"
        select="replace($text,
''(s?t?(&quot;|\s|[.,;:]|(balmed|dention|em|gainst|ll|mong|parm
|liquid|specially|spose|stead|teak|till|[Tt]is|[Tt]was)(,|\.\s)|$
))', '&rsquo;$1')"
        as="xs:string" />
    <xsl:variable
        name="text"
        select="replace($text, ' (^|\s|&quot;|-)' ( [&quot;]|$) ',
'$1&lsquo;$2')"
        as="xs:string" />
    <xsl:variable
        name="text"
        select="replace($text, ' (^|-|\s)&quot;;', '$1&ldquo;')"
        as="xs:string" />
    <xsl:variable
        name="text"
        select="replace($text, '&quot;(\s|[-.,;:]|$) ',
'&rdquo;$1')"
        as="xs:string" />
    <xsl:variable
        name="text"
        select="replace($text, ' ([^\s])' ' ([^\s]) ', '$1&rsquo;$2')"
        as="xs:string" />

    <!-- Variations on '* * *' in 'Extracts'. -->
    <xsl:variable
        name="text"
        select="replace($text, ' \*', '&#xA0;&#xA0;*)'"
        as="xs:string" />
    <xsl:variable
        name="text"
        select="replace($text, '\* ', '*&#xA0;&#xA0;')"
        as="xs:string" />

    <xsl:sequence select="$text" />
</xsl:function>
```



‘ahf:text()’ XSLT 関数は他の文脈でも使用されます。たとえば、次のとおりです。

```
<xsl:template match="div[@type = 'fly_title']/bibl">
  <fo:block text-align="center" hyphenate="false" font-size="5pt"
    line-height="10.5pt"
    space-before="2.33in"
    space-before.conditionality="retain">
    <!-- Provide structure that is not in the source XML. -->
    <xsl:analyze-string select="ahf:text(edition/text())"
      regex="HERMAN MELVILLE,">
```

... em dashes

### 連続した em ダッシュ

初版では、複数の場所でタイポグラフィック効果として ふたつまたは 三つの連続した em ダッシュが使用されています。次に例を示します。

----- “There Leviathan,  
Hugest of living creatures, in the deep

em ダッシュについて解説しているタイポグラフィの本のほとんどは、ダッシュの前後に細い空白を入れることを推奨しています。例えば、「Correct Composition (8)」では次のように述べられています。

*>As the dash entirely fills the body sideways, it should have before and after it a thin space to prevent the interference with adjoining characters.*

ダッシュは文字の横幅いっぱいに広がるため、隣接する文字との干渉を防ぐために、ダッシュの前後に細い空白を設ける必要があります。

多くのデジタルフォントは、em ダッシュが幅を完全に埋めるという活版印刷の慣習を守っています。ただし、Source Serif Pro には、ストロークの前後に組み込まれた空白が含まれています。これは一般的に便利ですが、連続した em ダッシュがある場合は見栄えが悪くなります。

--“There Leviathan,  
Hugest of living creatures, in the deep

連続する em ダッシュを fo:wrapper font-family="serif" で囲んで、結合する em ダッシュのフォントを選択する必要があると思われていました。しかし、連続する ふたつ、または三つの em ダッシュの Unicode 文字が偶然 (再) 発見され、フォントを変更せずに正しいダッシュを表示する方法が見つかりました。テキスト処理にさらに手順が追加されました。

```
<xsl:variable
  name="text"
  select="replace($text, '&mdash;&mdash;&mdash;', '&#x2E3B;')"
  as="xs:string" />
<xsl:variable
  name="text"
  select="replace($text, '&mdash;&mdash;', '&#x2E3A;')"
  as="xs:string" />
```

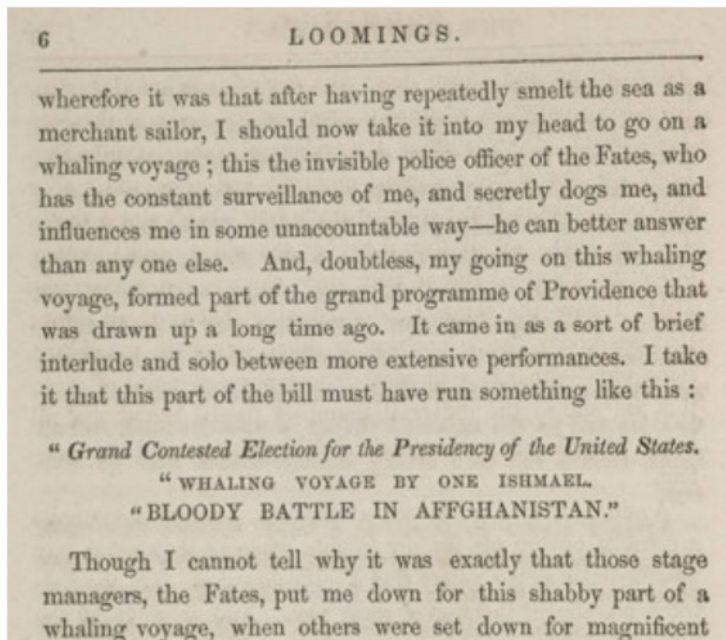
同じフォントを使って正しいダッシュを使用できるようになりました。

—“There Leviathan,  
Hugest of living creatures, in the deep

## 行グリッド

ページの裏側のテキストが紙を通して見える場合があります。裏側のテキストの影により、表側のテキストの読みやすさが低下します。これを防ぐひとつの方法は(厚い紙を使用するか、文書を電子的にのみ読むこと以外)、ページの表側と裏側のテキストの行を揃えることです。

初版のこの画像には、多少のこの現象が見られますが、線がほぼ揃っていることと、3本の不規則な線の後に線が再び揃っていることも示されています。

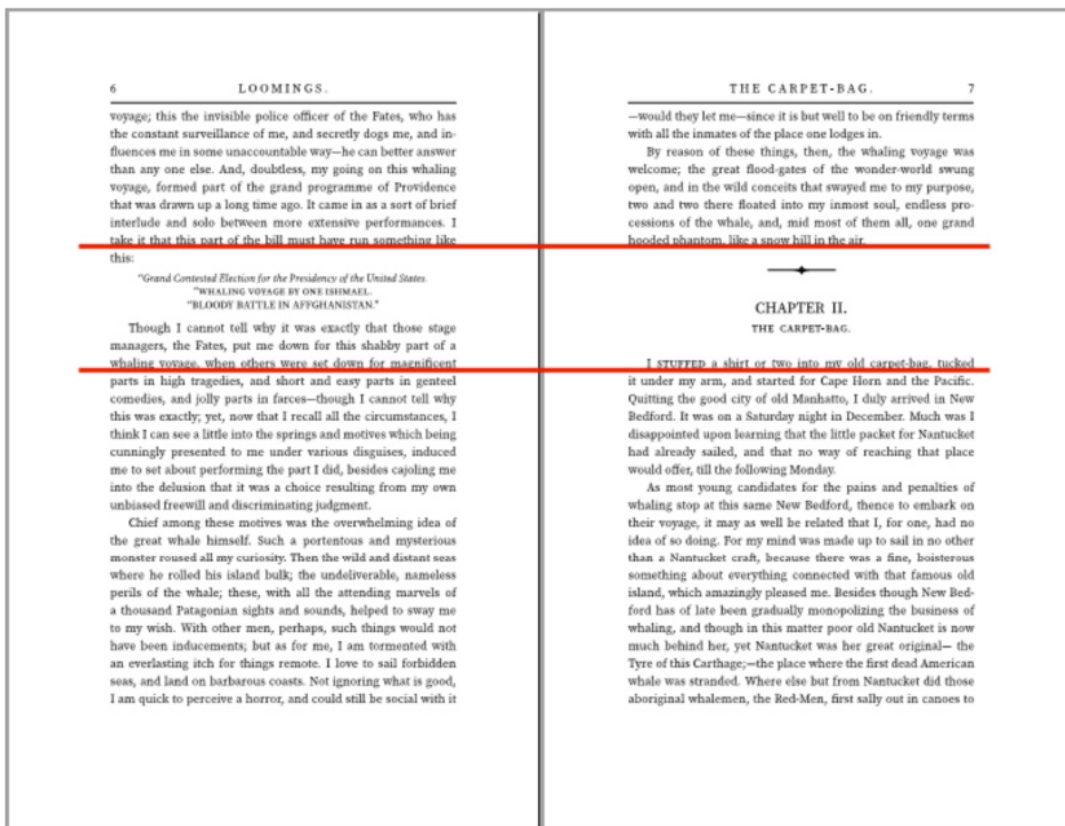


すべてのテキストのフォントサイズと行の高さが同じであれば、行の前後を揃えるのは簡単です。テキストにタイトルなどが含まれており、フォントサイズ、行の高さ、前後のスペースが異なる場合は、難しくなります。タイトルの前の空白、タイトルの行の高さ、タイトルの後の空白の合計が基本行の高さの倍数になるようにタイトルの

## Formatting Moby-Dick

スタイルを設定することは、多くの場合可能です。ただし、一部のタイトルが2行にまたがり、タイトルの行の高さが基本行の高さの倍数でない場合は、この方法は失敗します。

Antenna House Formatter の baseline grid 拡張機能は、行を共通のベースラインに揃えるだけでなく、特定のブロック内の行を独自のグリッドに揃えたり、グリッドに揃えないようにしたりすることもできます。次の図の赤い線は、通常の段落内の行が、三つの不規則な行の後や章番号とタイトルの後でも行グリッドに揃えられていることを示しています。



最初のステップは、「axf:baseline-grid」を使用して行グリッドを指定することです。

```
<xsl:template match="body">
  <fo:page-sequence
    master-reference="PageMaster"
    writing-mode="from-page-master-region()"
    initial-page-number="1"
    axf:baseline-grid="root">
    <xsl:call-template name="PageMaster-static-content" />
    <fo:flow flow-name="xsl-region-body" hyphenate="true"
      text-align="justify">
      <xsl:apply-templates />
    </fo:flow>
  </fo:page-sequence>
</xsl:template>
```

```
    </fo:flow>
  </fo:page-sequence>
</xsl:template>
```

次のステップは、やはり `axf:baseline-grid` を使用して行グリッドを使用しないブロックに独自のグリッドを確立させることです。

```
<xsl:template match="body//q">
  <fo:block text-align="center"
    text-indent="0"
    space-before="0.25lh"
    font-size="7pt"
    line-height="9pt"
    axf:baseline-block-snap="before margin-box"
    axf:baseline-grid="new">
    <xsl:apply-templates />
  </fo:block>
</xsl:template>
```

`axf:baseline-block-snap` は、ブロックを行グリッドにどのように揃えるかを指定します。

## ヘッダとフッタ

初版のヘッダとフッタは、存在する場合、ページ番号と章のタイトルのみという非常にシンプルなものでした。ただし、長いタイトルのない章でも、一部の章では省略されたタイトルが使用されています。TEI XML にはランニングヘッダのテキストが含まれていないため、省略されたタイトルは `<fw>` (「forme work」) 要素として追加されました (8)。次に例を示します。

```
<div type="chapter">
  <head>CHAPTER XXIX.</head>
  <head type="sub">ENTER AHAB; TO HIM, STUBB.</head>
  <fw type="head" place="top-centre">ENTER AHAB.</fw>
```

ランニングヘッダの `<fo:marker>` のコンテンツとして、タイトルテキストよりも `<fw>` 要素 (存在する場合) を選択するのは簡単です。

```
<fo:marker marker-class-name="Chapter-Title">
  <xsl:apply-templates
    select="(fw[@type = 'head'], head) [1]/node()"
    mode="marker" />
</fo:marker>
```

略したタイトルは通常、ヘッダの中央に配置されます。

THE VIRGIN. <span style="float: right;">401</span>
<hr/> "don't be in such a devil of a hurry to sink! By thunder,

ただし、略したタイトルでもかなり長くなる場合があります。少なくともひとつのタイトルは、ページ番号を圧迫せずにヘッダの中央に配置することができないほど長いです。

THE HONOR AND GLORY OF WHALING. 405
<hr/> up by a whale; still, whether that strictly makes a whaleman

解決策としては、ヘッダが大きすぎる場合はオーバーフローさせ、ページ番号が本文領域の外側の端に揃うように `axf:overflow-align` を指定することです。

```
<xsl:template name="Odd-Header">
  <fo:block
    keep-together.within-line="always"
    text-align="center"
    font-size="8pt"
    border-bottom="1pt solid black"
    axf:leader-expansion="force"
    padding-bottom="5pt"
    margin-bottom="4pt"
    axf:overflow-align="end">
    <fo:page-number color="transparent"/>
    <fo:leader />
    <fo:inline letter-spacing="0.22em">
      <fo:retrieve-marker
        retrieve-class-name="Chapter-Title"
        retrieve-position="last-starting-within-page" />
    </fo:inline>
    <fo:leader />
    <fo:page-number />
  </fo:block>
</xsl:template>
```

THE HONOR AND GLORY OF WHALING. 405
<hr/> brawny doer of rejoicing good deeds, was swallowed down

## おわりに

ハーマン・メルヴィルの Moby-Dick 『白鯨』 のアメリカ初版を組版するためのスタイルシートの開発にはいくつかの課題がありましたが、XSLT、XSL-FO、および Antenna House Formatter 拡張機能を組み合わせて使用することで、これらの課題を解決することができました。

## 参考文献

1. **IU Digital Library Program.** Moby-Dick, or, The Whale. Melville, Herman, (1819–1891). [Online]  
<http://web.archive.org/web/20200815114219/http://webapp1.dlib.indiana.edu/TEIgeneral/view?docId=wright/VAC7237&brand=wright>.
2. —. Moby Dick, or, The Whale. [Online]  
<http://web.archive.org/web/20201027153917/http://dogwood.dlib.indiana.edu:8080/xubmit/rest/repository/wright/VAC7237.xml>.
3. —. Wright American Fiction. *Indiana University*. [Online]  
<http://web.archive.org/web/20200725213048/http://webapp1.dlib.indiana.edu/TEIgeneral/welcome.do?brand=wright>.
4. **Antenna House.** Automated Analysis. *AH Formatter V7.0*. [Online]  
<https://www.antenna.co.jp/AHF/help/v70e/ahf-analyzer.html>.
5. —. Antenna House Formatter V7. [Online] <https://www.antennahouse.com/formatter-v7>.
6. **Grießhammer, Frank.** Introducing Source Serif 2.0. *Adobe Typekit Blog*. [Online] January 10, 2017. <https://blog.typekit.com/2017/01/10/introducing-source-serif-2-0/>.
7. **Wikipedia.** Moby-Dick. *Wikipedia*. [Online] [https://en.wikipedia.org/wiki/Moby-Dick#Chapter\\_structure](https://en.wikipedia.org/wiki/Moby-Dick#Chapter_structure).
8. **Text Encoding Initiative.** Headers, Footers, and Similar Matter. *P5: Guidelines for Electronic Text Encoding and Interchange*. [Online] <https://tei-c.org/release/doc/tei-p5-doc/en/html/PH.html#PHSK>.