もうAcrObatなんていらない！？

PDFを自分で作ってみよう～♪

BY IIJIMAS

アジェンダ

- PDFとは何か・PDFの作成方法
- PDFのファイル構造
- PDFの文法・要素・文書構造
- PDFのグラフィックス・テキストの描画

PDFとは何か<そんなのみんな知ってるだろうけど…

- Adobe Systems社によって開発された、電子文書のためのフォーマット
  Portable **D**ocument **F**ormat
  - 印刷ページ記述言語PostScriptが原型になっている
- Adobe Reader等でPDF文書の閲覧できる
- ファイルフォーマットの仕様は公開されている
  - http://www.adobe.com/devnet/pdf/pdf_reference.html
    - 最新版のPDFリファレンス(PDF1.7, Acrobat 8,英語)
      - » http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf
    - 日本語版(PDFリファレンス第2版—Adobe Portable Document Format Version 1.3)
      - » http://www.amazon.co.jp/dp/4894713381

PDFの作成方法（ http://ja.wikipedia.org/wiki/PDFの方が詳しいw）

- Acrobatを使う→購入する必要がある!
- **2007 Microsoft Office プログラム用 Microsoft PDF/XPS 保存アドイン →2007 Officeが必要**
  - http://www.microsoft.com/downloads/details.aspx?FamilyID=4d951911-3e7e-4ae6-b059-a2e79ed87041&displaylang=ja
- その他
  http://ja.wikipedia.org/wiki/PDF%E3%82%BD%E3%83%95%E3%83%88%E3%82%A6%E3%82%A7%E3%82%A2%E3%81%AE%E4%B8%80%E8%A6%A7
  - 商用ソフトウェア、オープンソースソフトウェア(OpenOffice.org, GhostScript,･･･等など)、フリーウェア(クセロPDF, PrimoPDF等)
  - ライブラリ(iText, PDFLib･･･)

PDFの仕様は公開されているので…

- PDFリファレンス(PDF Reference, version 1.7)
  http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference_1-7.pdf
  があれば誰でも作成できるはず！
- 実は基本はテキストベースです。
- ~~暇人~~プログラマなら一度はPDFの内部を解析してみたいと思いませんか！
- 自分で作成してみましょう！
  - 以下に最小のサンプルがありますが…
    - PDF Reference, version 1.7(p.1057)
      - » APPENDIX G - GExample PDF Files-G.1Minimal PDF File

PDFのファイル構造

- ヘッダ
  - %PDF-1.4
    - バイナリを含む場合この後に4bytes以上のバイナリ文字推奨
      ファイル転送ソフトにバイナリファイルとして認識させるため。
- ボディ
  - 間接オブジェクト(詳細後述)の並び
    - (オブジェクト参照番号) (生成番号) obj ～ endobj
      1 0 obj
      （ここに内容 詳しくは後述）
      endobj

PDFのファイル構造

- 相互参照表
  - 各間接オブジェクトのオフセットの表
    - xref (サブセクション1) (サブセクション2) …
    - サブセクション（通常1つでよい）
      - (先頭オブジェクト番号) (エントリ数)
      - エントリnnnnnnnnnn ggggg (nかf) (eol) 計20 bytes
        例：0000000010 00000 n
- トレーラ
    - trailer (トレーラ辞書Rootのオフセットなど詳細後述)
    - startxref (xrefのオフセット、Readerはここから読む)
    - %%EOF

最小限のPDFファイルのサンプル

ヘッダ

ボディ

```
%PDF-1.4
1 0 obj
<</Type /Catalog
/Pages 2 0 R
>>
endobj
2 0 obj
<</Type /Pages
/Kids [3 0 R]
/Count 1
>>
endobj
3 0 obj
<</Type /Page
/Parent 2 0 R
/MediaBox [0 0 612 792 ]
/Contents 4 0 R
/Resources <</ProcSet 5 0 R >>
>>
endobj
4 0 obj
<</Length 0 >>
stream

endstream
endobj
```

相互参照表

トレーラ

```
5 0 obj
[/PDF ]
endobj
xref
0 6
0000000000 65535 f
0000000010 00000 n
0000000063 00000 n
0000000125 00000 n
0000000251 00000 n
0000000305 00000 n
trailer
<< /Size 6
/Root 1 0 R
>>
startxref
331
%%EOF
```

まっさらな「A4白紙1ページ」のPDFです。
まだ小さくできるはずです！

PDFの文法

- 文字
  - 通常文字（下記以外）
  - 区切り文字 (,),<,>,[,],{,},/,%
  - 空白文字（連続していても1つとして扱われる）
    0x00 (NULL),0x09,(TAB) 0x0A (LF),0x0D(CR),
    0x20(SPACE)….
- コメント　% から行末まで
  - ヘッダ%PDF-1.4と%%EOF以外意味はない。

PDFの要素

- オブジェクト（PDFの基本要素）

オブジェクト	説明	例
ヌル	なしを意味する	null
論理値	true / false	true false
整数	符号付き整数値	0 -10 100
実数	符号付き実数値	0.2 2 -1.1 .003
文字列	文字列	(abc) <FEFF006100620063>
名前	一意の識別子	/AAA /BBB
配列	1次元のコレクション	[0 0 200 100] [[1 0] [0 1]]
辞書	名前-オブジェクトの連想配列	<</A 1 /B 2 /C 3>>
ストリーム	バイト列	辞書+stream ～endstream

PDFの要素

- 文字列オブジェクト
  - 文字列をあらわすバイト列
  - リテラル表記
    - ()で囲む
    - 例:(ABC)
    - 一部の文字はエスケープする必要あり
  - 16進表記
    - <>で囲む
    - 文字コードを16進数で表記
    - 例:<FEFF006100620063>

リテラル文字列エスケープシーケンス

	意味
¥n	改行(LF)
¥r	復帰(CR)
¥t	タブ
¥b	バックスペース
¥f	改頁
¥(	左括弧
¥)	右括弧
¥¥	¥

PDFの要素

- 辞書オブジェクト
  - PDFの主要な構成要素
  - <<と>>に囲まれたキーと値オブジェクトの並び
    - <<
      /Key1 Value1
      /Key2 Value2
      …..
      >>
  - キーは名前オブジェクトである必要がある
  - Type項目によって型を特定する(値は名前オブジェクト)

PDFの要素

- ストリームオブジェクト
  - バイト列を表す（画像やページコンテンツに使用）
    - (ストリーム辞書)
      stream
      (バイトの並び)
      endstream
  - ストリーム辞書
    - 主な項目

キー	値の型	説明
Length	整数	バイト列の長さ
Filter	名前、配列	バイト列に適用されるフィルタ種類

PDFの要素

- 間接オブジェクト
  - オブジェクト識別子
    - オブジェクト番号
    - 生成番号 …新しいファイルでは0
  - 間接オブジェクトの定義
    - (オブジェクト番号) (生成番号) **obj**
      (内容)
      endobj
  - 間接オブジェクトの参照
    - (オブジェクト番号) (生成番号) **R**

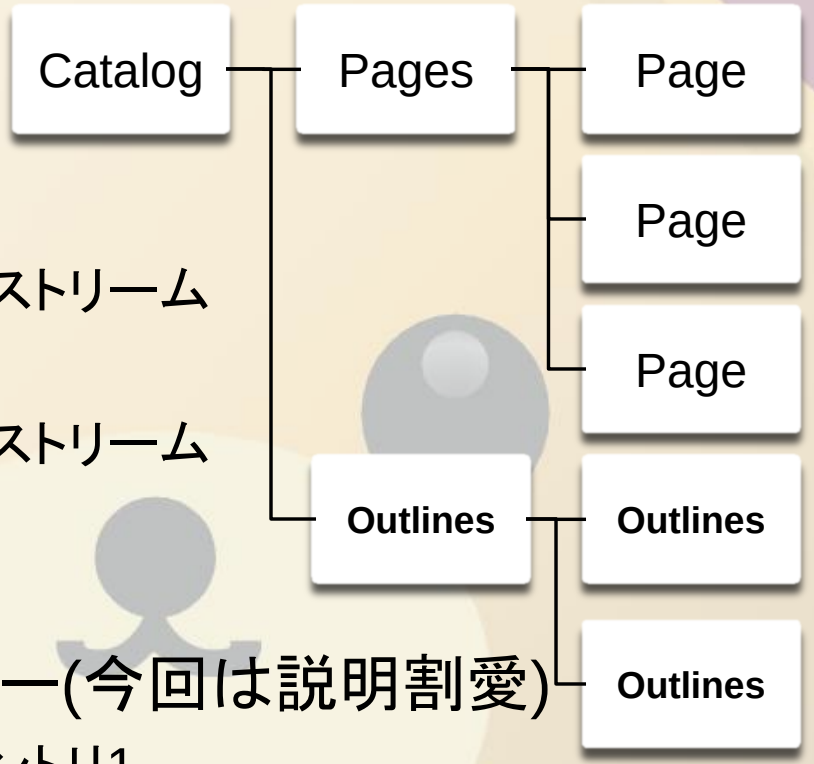
PDFのファイル構造2

- trailer辞書
  - 文書レベルでの特殊なオブジェクトの情報
    - PDFを読み込むアプリのために存在する

キー	値の型	説明
Size	整数	相互参照表に含まれるエントリの数
Root	辞書	Catalog辞書
Encrypt	辞書	暗号化辞書
Info	辞書	文書情報辞書
ID	配列	ファイル識別子

PDFの文書構造

- 文書カタログ
  - ページツリー
    - ページ1
      - » コンテントストリーム
    - ページ2
      - » コンテントストリーム
    - ….
  - アウトラインツリー(今回は説明割愛)
    - アウトラインエントリ1
    - ….



PDFの文書構造

- 文書カタログ
  - Catalog辞書
    - PDF文書のルート。
    - Trailer辞書のRootの値となる
    - 主な項目

キー	値の型	説明
Type	名前	Catalog
Pages	辞書	ページツリーのルート
Outlines	辞書	しおりツリーのルート

PDFの文書構造

- ページツリー
  - ページノード辞書
    - ページツリーのノード
    - 主な項目

キー	値の型	説明
Type	名前	Pages
Parent	辞書	親Pages辞書
Kids	配列	PagesまたはPage辞書の配列
Count	整数	子孫Pageの数

PDFの文書構造

- ページツリー(ページ)
  - ページオブジェクト辞書
    - ページツリーのリーフ
    - 主な項目

キー	値の型	説明
Type	名前	Page
Parent	辞書	親Pages辞書
Resources	辞書	コンテントストリームの中で参照するリソースの辞書
MediaBox	矩形(配列)	印刷時の出力可能最大領域
Contents	ストリーム	内容を記述するコンテントストリーム

PDFの文書構造

- ページ内容
  - リソース辞書
    - ページコンテントストリームから使用する文書レベルのオブジェクトの参照

キー	値の型	説明
ProcSet	配列	手続きセット（PostScript時代の名残）
Font	辞書	コンテントストリームの中で参照するリソースの辞書
Pattern	辞書	塗りつぶし模様
Sading	辞書	塗りつぶし模様
ColorSpace	辞書	色空間の指定

  - コンテントストリーム
    - ストリームオブジェクトである
    - ページに描画するグラフィックスオペレータから構成される(次のスライド以降で説明)
    - グラフィックスオブジェクト
      - パス、テキストからなるベクタグラフィックスが基本
      - ラスタイメージ
      - フォント
      - 塗りつぶしパターン

PDFグラフィックス

- グラフィックスの要素
  - パスと呼ばれる、直線とベジエ曲線で構成される図形が基本
- グラフィックスオブジェクト
  - パスオブジェクト
  - テキストオブジェクト
  - インラインイメージ
- グラフィックスオペレータ(※大・小文字区別あり)
    - (オペランドの並び) (オペレータ)

PDFグラフィックス

- パス構築オペレータ
  - パスオブジェクトを作成する。

オペランド	オペレータ	説明
x y	m	パスを開始 カレントポイントを(x,y)に移動する いわゆるMoveTo
x y	l	カレントポイントを始点、(x,y)を終点とする線分を描き、カレントポイントを(x,y)に移動 いわゆるLineTo
x1 y1 x2 y2 x3 y3	c	3次Bezier曲線を描画
	h	Pathを閉じる(カレント-始点を結ぶ)
x y w h	re	左下座標(x, y)幅w高さhの矩形

PDFグラフィックス

- パスペイントオペレータ
  - パスを終了し、ペイントする。

オペランド	オペレータ	説明
	s	ストロークする。(輪郭を描く)
	f	パスの内部を塗りつぶす。
	b	パスの内部を塗りつぶしストロークする。
	f*	fと同じ。塗りつぶしが**ALTERNATE**モード
	b*	bと同じ。塗りつぶしが**ALTERNATE**モード
	n	(パスを終了するだけ)

PDFグラフィックス

- グラフィックス状態オペレータ
  - 状態を操作するもの

オペランド	オペレータ	説明
	q	カレントグラフィックス状態をスタックに保存 SaveDCのようなもの
	Q	カレントグラフィックス状態をスタックから復元 RestoreDCのようなもの
a b c d e f	cm	カレント座標変換行列に行列を連結
lineWidth	w	線幅
LineCap	J	線の端の形状(0,1:ラウンド,2:スクウェア)
lineJoin	j	線の結合部の形状(0,1:ラウンド,2:ベベル)

PDFグラフィックス

- カラーオペレータ

オペランド	オペレータ	説明
r g b	RG	ストロークの色をセットする(0.0-1.0)
r g b	rg	塗りつぶしの色をセットする(0.0-1.0)
gray	G	ストロークの色をセットする(0.0-1.0)
gray	g	塗りつぶしの色をセットする(0.0-1.0)
c m y k	K	ストロークの色をセットする(0.0-1.0)
c m y k	k	塗りつぶしの色をセットする(0.0-1.0)
c1 c2 c3 name	SCN	ストロークの色空間の設定
c1 c2 c3 name	scn	塗りつぶしの色空間の設定

PDFグラフィックス

- イメージ
  - イメージXObject(今回は残念ながら割愛)
  - インラインイメージ（4KB以下推奨）

オペレータ	説明
BI	インラインイメージオブジェクトを開始
ID	イメージデータの開始
EI	インラインイメージオブジェクトを終了

詳細略。
例:

BI
/W 16 %(幅)
/H 16 %(高さ)
/BPC 8 %(bit深度)
/D [0 1] %(bit→色)
ID (データ)
EI

PDFのテキストの描画

- テキストオブジェクトオペレータ

オペレータ	説明
BT	テキストオブジェクトを開始
ET	テキストオブジェクトを終了

- テキスト配置オペレータ

オペランド	オペレータ	説明
x y	Td	カレントを相対座標(x ,y)に移動する
a b c d e f	Tm	テキスト行列を指定のものに置き換える

- テキスト表示オペレータ

オペランド	オペレータ	説明
string	Tj	Stringを表示

- テキスト状態オペレータ

オペランド	オペレータ	説明
font size	Tf	カレントフォントとカレントフォントサイズを設定
charSpace	Tc	文字間スペーシング
scale	Tz	水平スケーリング

PDFのフォント

- フォントデータ
  - フォント辞書で定義される
  - いくつかのフォントタイプに分類されSubtype項目で指定される

Subtype	説明
Type0	コンポジットフォント
Type1	通常の欧米圏フォント
TrueType	TrueTypeに基づく
CIDFontType2	Type0の子になるフォント TrueTypeに基づく 日本語フォントはすべてこれ

PDFのフォント

- **フォント辞書**(とても詳細にご紹介しきれないです…)
- 例だけで許してください…
  - 小さいですが…

Type1フォント「Helvetica(Arial)」の例:

```
<<
/Type /Font
/Subtype /Type1
/BaseFont
/Helvetica
/Encoding
/WinAnsiEncoding
>>
```

Type0(CIDFontType2)フォント「MS Pゴシック」の例:

```
7 0 obj
<<
/Type /Font
/Subtype /Type0
/BaseFont /MSPGothic
/Encoding /90msp-RKSJ-H
/DescendantFonts [8 0 R]
>>
endobj
8 0 obj
<<
/Type /Font
/Subtype /CIDFontType2
/BaseFont /MSPGothic
/FontDescriptor 9 0 R
/CIDSystemInfo << /Registry (Adobe) /Ordering (Japan1)
/Supplement 2 >>
/W 10 0 R % 文字幅配列の参照(詳細略)
/DW 1000
>>
```

% FontDescriptor辞書

```
9 0 obj
<<
/Type /FontDescriptor
/Ascent 859
/CapHeight 859
/Descent -141
/Flags 6
/FontBBox [-100 -141 842 1000]
/FontName /MSPGothic
/ItalicAngle 0
/StemV 76
/XHeight 430
/StemH 76
/MissingWidth 418
/MaxWidth 742
/AvgWidth 418
/Style <</Panose
<000000000000000000000000>>>
>>
```

まとめ・結論（？）

- PDFファイルは主に辞書オブジェクトとストリームオブジェクトからなる。
- 目に見える部分はグラフィックスオペレータで描画されている。
- ご紹介しきれなかったことも多数あります。
  - ごめんなさい、~~無駄な努力~~茨の道かもしれません…
  - 本格的な画像挿入や日本語の表示をするとなると…かなり面倒です（笑）
  - というわけで、実用的にはiTextなどのライブラリの使用を推奨します(笑)