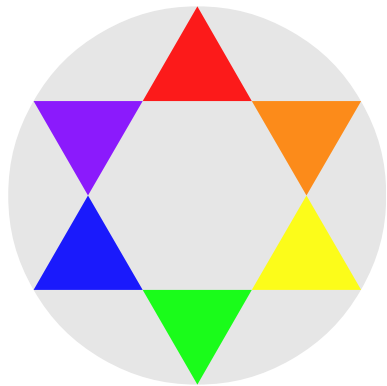


Introduction of Python ezdxf Module for AutoCAD DXF File



妹尾 賢 (SENOO, Ken)

✉ contact@senooken.jp

🗣 <https://social.senooken.jp/senooken>

2015-03-14

第 36 回 オープン CAE 勉強会@関西

<http://ocbkansai.connpass.com/event/12007/>

URL: <https://senooken.jp/public/20150314/>



This work is licensed under the [CC0 1.0 Universal License](https://creativecommons.org/licenses/by/4.0/). To the extent possible under law, I have waived all copyright and related or neighboring rights to this work.

目次

- 背景
- DXF ライブラリの調査
 - Python
 - Ruby , C/C++
- CAD データ配布サイト
- ezdxf のお試し
 - サンプル
 - 実行
- まとめ

DXF ライブラリ調査の背景

計算対象領域（モデル）の作成手順

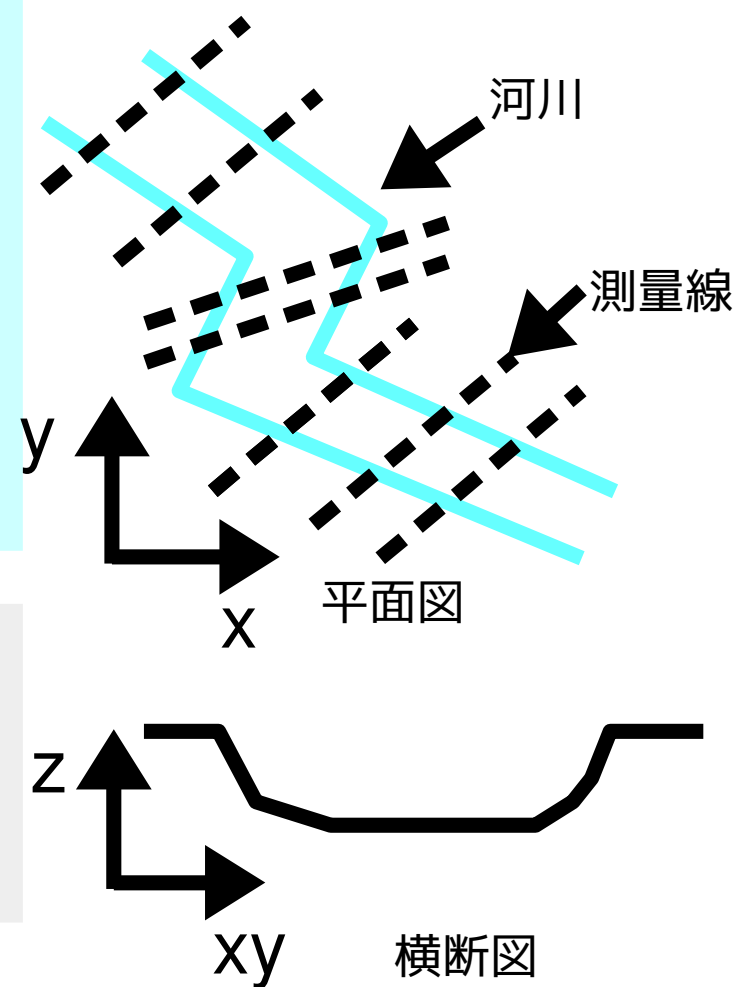
- 平面図から輪郭をトレース▷ xy 分布
- 横断面図をトレース▷ z 分布
- xy に z を補間▷ 計算メッシュ

平面図：

上空からみた地形図。地形の平面分布を確認可能。

横断面図：

測量線の位置で横から見た地形図。起伏を確認可能。



- 平面図と横断面図の CAD からの**手作業トレース**必須
- **領域が広く数が多い**とやってられない

使用 CAD データの種類

DXF (Drawing Exchange Format)

- **テキスト**形式
- **オープン**な仕様

http://images.autodesk.com/adsk/files/acad_dxf0.pdf

- データサイズ：**大**
- CAD 図面のデファクトスタンダード

DWG

- **バイナリ**形式
- **クローズド**な仕様

- データサイズ：**小**
- AutoCAD 専用の図面データのファイル形式

他には JW_CAD で
使われる jww 形式もある

- ・ 紙図面 (pdf) しかなくともトレースして DXF に変換
- ・ DXF は**テキスト**で**仕様がオープン**なので構文解析が簡単
- ・ **DXF で図面を保存しライブラリでデータにアクセス**

DXF ライブラリ

ライブラリの選定

- Python を念頭に、参考として以下 3 言語を調査

- Python

- ▶ 自分が見える。
- ▶ 科学技術計算での利用が多くライブラリの存在期待。

- Ruby

- ▶ Python と立ち位置が似ている。
- ▶ 比較参考情報として状況を調査。

- C/C++

- ▶ 速度重視のときに使う可能性あり。

Python の DXF ライブラリ

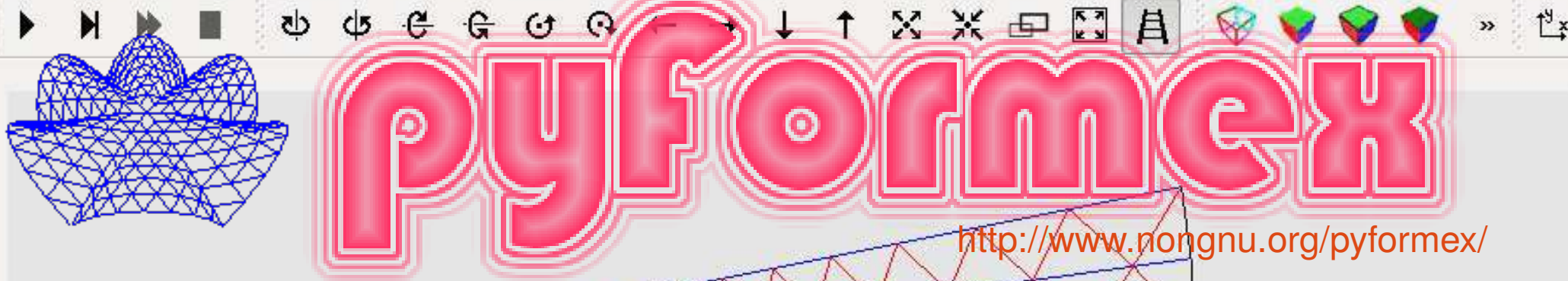
名前	バージョン	更新日付	説明	作者	ライセンス	URL
dxfgripper	0.7.4	2014-05-23	読み込みのみ対応。	Manfred Moitzi	MIT	https://bitbucket.org/mozman/dxfgripper
dxfwrite	1.2.0	2012-08-11	DXF R12の書き込専用。	Manfred Moitzi	MIT	https://bitbucket.org/mozman/dxfwrite
ezdxf	0.6.5	2015-02-27	読み書き可能。	Manfred Moitzi	MIT	https://bitbucket.org/mozman/ezdxf
pyFormex	0.9.1	2013-10-15	3Dモデルの作成, FEAのプリポスト。GUIプログラム。DXFの読み書きも可能。	Benedict Verheghe	GPL3	http://www.nongnu.org/pyformex/
SDXF	1.1.1	2012-07-00	書き出し専用。	Stani	不明	https://github.com/nycresistor/SDXF

dxfgripper , dxfwrite , ezdxf は同一作者。 ezdxf に開発成果を集中？

読み書きライブラリは **ezdxf** 一択

参考： My Future Sight for Past: Python の CAD(.dxf) ファイル読み書きモジュールまとめ

<http://myfuturesightforpast.blogspot.jp/2014/07/pythoncaddxf.html>



- * 巨大な 3D 構造物のモデルの操作・生成・加工。
- * Python のスクリプトで処理可能。
- * STL や FEA のメッシュャー, CFD 格子で活用。
- * Gmsh 的なソフト？
- * DXF の読み書きにはパーサーの指定が必要。

Currently, quad8 and quad9 elements can be drawn as quadratics in the smooth or flat rendering style. To activate the quadratic surface drawing, change the default in Settings->Drawing.

Ruby の DXF ライブラリ（参考）

名前	バージョン	更新日付	説明	作者	ライセンス	URL
DXF	0.3	2014-11-19	読み書き。	Brandon Fosdick	BSD	https://github.com/bfoz/dxf-ruby
ruby-dxf-reader	1.1	2011-06-22	読み書き。	Jim Foltz	LGPL2.1	https://github.com/jimfoltz/ruby-dxf-reader
dxf-in-ruby	0.0.1	2014-03-13	読み書き。	Brandon Fosdick, Meseker Yohannes	不明	https://www.versioneye.com/ruby/dxf-in-ruby/

- * Python も Ruby も **DXF ライブラリ**は少ない。
- * 日本語でも英語でも**情報**は少ない。
- * Python の **ezdxf** のほうが開発は盛んな印象。

その他のライブラリ

- LibreDWG （ C ：開発中（停止） ） : <http://www.gnu.org/software/libredwg/>
- dxfllib （ C++ ） : <http://www.qcad.org/en/what-is-dxfllib>

CAD データ

動作確認用 CAD データの用意

- ezdxf の動作確認のため CAD データを用意
- 自作（自由な CAD ソフト）
 - JW_CAD ：建築系ではメジャーらしい。
 - LibreCAD ：個人的に 2D CAD ではベスト。
 - データ・形状確認などで利用。
- 配布サイトからダウンロード
 - 家電や建築材，メーカー企業で配布
 - ただし，個人利用のみの制限
 - 緩いライセンスの配布サイト
- GIS など他の形式から変換

CAD データの配布サイト

サイト名	内容	2 次・再配布	営 利	コメント	URL
nousera.com	建築CADデータ・点景データ。	×	?	利用規則条文が少ない。	http://www.nousera.com
CAD-data.com	CAD データ共有サービス。	×	○	データの投稿者に責任があり使いやすい。	http://www.cad-data.com/
無料 CAD データサイトまとめ	Naver まとめ。	?	?	データが自由とは限らないがよくまとまってる。	http://matome.naver.jp/odai/2131978562759949901
車の塗り絵・CAD データ	車の塗り絵とフリーの CAD データ。	×	?	CAD : JWW と DXF 形式、塗り絵: PDF 形式。	http://nurie-cad.matrix.jp/
kibousya/ 機械設計資料	JIS 企画のネジなど。	?	?	利用規則条文が少ない。	http://www.geocities.jp/meg eo1/cadatamenu.html
土木資料館	土木関係の物品。	?	○	DXF と JWW 形式。	http://www.sakai11.jp/index.htm
豊中市:デジタル地図データの提供	デジタル地図データサンプル。	?	?	地図の DXF データのサンプル。	https://www.city.toyonaka.osaka.jp/machi/doro/daityou/digitalmap/index.html
COM-ET	主要 TOTO 商品の CAD データ。	×	×	2D と 3D CAD データ。個人利用のみ。	http://www.com-et.com/jp/page/cad/

CAD データの配布サイト

サイト名	内容	2次・再配布	営利	コメント	URL
nousera.com	建築CADデータ・点景データ。	×	?	利用規則条文が少ない。	http://www.nousera.com
CAD-data.com	CAD データ共有サービス。	×	○	データの投稿者に責任があり使いやすい。	http://www.cad-data.com/
無料 CAD データサイトまとめ	Naver まとめ。	?	?	データが自由とされているがよくまとまっている。	http://www.geocities.jp/5994990
車の塗り絵・CAD データ	車の塗り絵とフリーの CAD データ。	×	?	CAD : JWW と式、塗り絵 : PD	http://www.geocities.jp/5994990
kibousya/ 機械設計資料	JIS 企画のネジなど。	?	?	利用規則条文が少ない。	http://www.geocities.jp/5994990
土木資料館	土木関係の物品。	?	○	DXF と JW	http://www.geocities.jp/5994990
豊中市:デジタル地図データの提供	デジタル地図データサンプル。	?	?	地図の DX	http://www.geocities.jp/5994990
COM-ET	主要 TOTO 商品の CAD データ。	×	×	2D と 3D CAD データ。個人利用のみ。	http://www.com-et.com/jp/page/cad/

ライセンスがゆるく
多様なデータが入手可能。

実製品の 3D CAD が
入手できるのは興味深い。

種類・数・制限から最も有力な配布サイト

パスワード:.....

ログイン

主な形式: dxf, dwg, jww

CAD-data.com はCADデータ共有サービスです。



ツイート

174

いいね!

264

g+1

23

利用規約要約

新着データ

■ 個人・法人を問わず、「登録ユーザー」が利用可能。

▶ 営利可能。

■ 2次配布・再配布は不可。

■ データ等の素材の責任は投稿者

■ 雑誌、図面への貼り付け・展示・提出可能

このサイトを利用するにはユーザー登録が必要です。

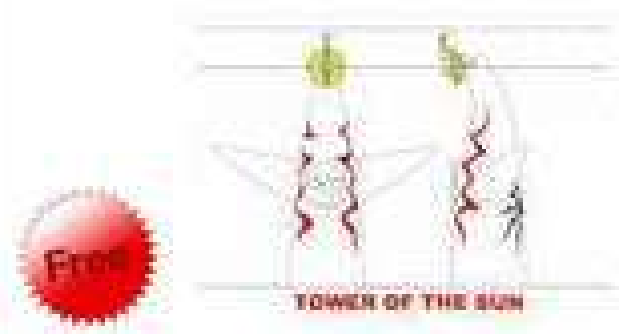
あなたが描いた自慢のCADデータハードディスクに眠っていませんか？
CAD-data.comにUPしてみんなと共有しましょう。

- ユーザー登録
- パスワードを忘れた

- タグ
- 投稿者リスト
- 新着データ

作業員 人 (作業高さ	飛行機スケルトン3	飛行機スケルトン2
イメージ用)	拡張子:jww	拡張子:jww
拡張子:dwg	拡張子:dwg	拡張子:dwg
CAD:AutoCAD	サイズ:1,046,836byte	サイズ:1,060,300byte
サイズ:33byte	次元:2D	次元:2D
更新日:2015-03-13	更新日:2015-03-12	更新日:2015-03-12
投稿者:304059さん	投稿者:mfmokaさん	投稿者:mfmokaさん
投稿回数:1回 (初)	投稿回数:45回	投稿回数:45回

面白かったデータ TOWER OF THE SUN



http://www.cad-data.com/cad/taneyon_35.html

黄金の顔

ドROID君



http://www.cad-data.com/cad/GRes_10.html

FENDER

http://www.cad-data.com/cad/taneyon_13.html



HONDAロゴ



http://www.cad-data.com/cad/ban16_6.html

ezdxf

インストールからチュートリアルの実行

ezdxf のマニュアル :

<http://ezdxf.readthedocs.org/en/latest/index.html>

インストール (Python2 でも 3 でも OK)

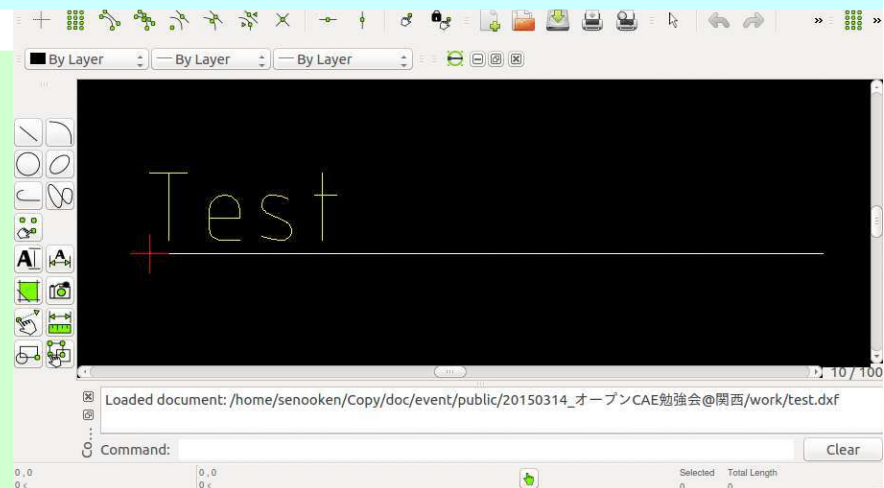
```
pip3 install ezdxf
```

サンプルの実行 ・ LibreCAD で表示

```
echo '#!/usr/bin/env python3
# coding: utf-8
# (File name: sample.py)
# https://bitbucket.org/mozman/ezdxf
```

```
import ezdxf
drawing = ezdxf.new(dxfversion="AC1024")
modelspace = drawing.modelspace()
modelspace.add_line((0, 0), (10, 0), dxfattribs={"color": 7})
drawing.layers.create("TEXTLAYER", dxfattribs={"color": 2})
modelspace.add_text("Test", dxfattribs={"insert": (0, 0.2), "layer": "TEXTLAYER"})
drawing.saveas("test.dxf")
' > sample.py
```

```
./sample.py
librecad test.dxf
```



基本概念

CAD ファイルに 3 種類のレイアウト

- model space : 構造物のデータ
- paper space : 印刷用データ
- block : ブロック要素

用語

- エンティティ (entity) : CAD で扱う図形要素。
点, 線分, ポリライン, テキスト, 寸法などを指す。

基本的には model space のエンティティを読み書き

基本操作（読み込み）

```
import ezdxf

FR = "./test.dxf"

dwg = ezdxf.readfile(FR)
msp = dwg.modelspace()

## 読み込みの手順は2通り
# 1. msp.query() メソッドで条件を指定してオブジェクトを取得
### ジオメトリがポリラインで画層が contour の
query = msp.query('LWPOLYLINE POLYLINE [layer="contour"]')

# 2. msp オブジェクトをまるごとループにかけて, オブジェクトの属性で判別
for entity in msp:
    if entity.dxf.layer == "contour" and
       (entity.dxftype() == "LWPOLYLINE" or entity.dxftype() == "POLYLINE"):
        pass
```

基本操作（書き込み）

```
import ezdxf

FW = "./polyline.dxf"
wdwg = ezdxf.new()
wmsp = wdwg.modelspace()

LAYER = "polyline"
dxfattribs = {"layer": LAYER}

wdwg.layers.create(LAYER)
points = [(0, 0), (1, 1)]

wmsp.add_polyline2d(points, dxfattribs=dxfattribs)

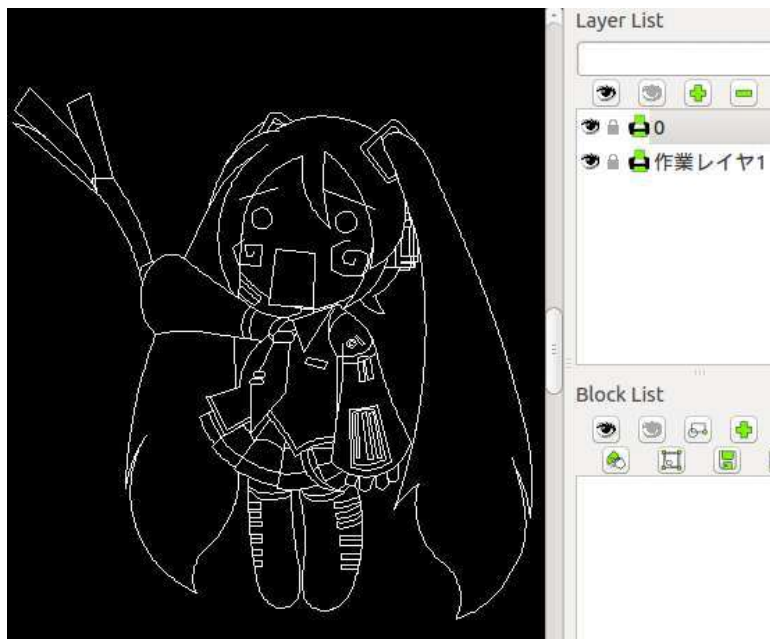
wdwg.saveas("polyline.dxf")
```

- 1)書き込みファイル開く
- 2)書式を設定
- 3)値を書き込む
- 4)ファイルを保存

実データを使った例

DL して LibreCAD で確認

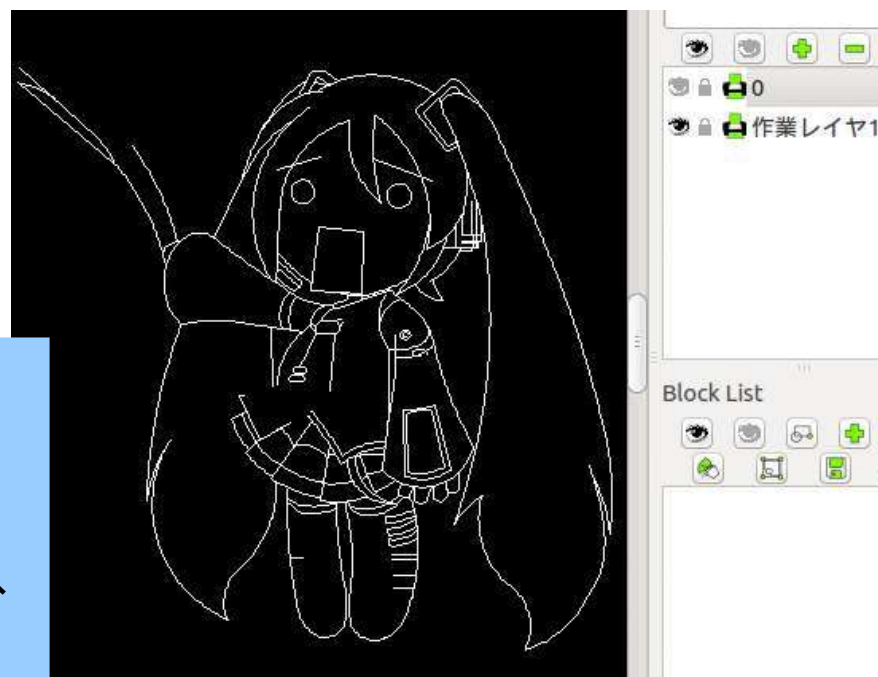
http://www.cad-data.com/cad/yasudasan_24.html



はちゅね三〇 CADで書いてみ



レイヤが2個
0 : えくぼとか
作業レイヤ1 : 大部分



ポリラインを抽出して出力する例

```
#!/usr/bin/env python3
# coding: utf-8
# (File name: extract.py)
```

```
import ezdxf
```

```
FR = "./yasudasan_24.dxf"
rdwg = ezdxf.readfile(FR)
rmisp = rdwg.modelspace()
```

```
query = rmisp.query('POLYLINE')
```

```
ex_line = []
for line in query:
    ex_line.append(list(line.points()))
```

```
FW = "./polyline.dxf"
wdwg = ezdxf.new()
wmisp = wdwg.modelspace()
```

```
LAYER = "polyline"
dxfattribs = {"layer": LAYER}
```

```
wdwg.layers.create(LAYER)
for line in ex_line:
    points = [point[:2] for point in line]
    wmisp.add_polyline2d(points, dxfattribs=dxfattribs)
```

```
wdwg.saveas("polyline.dxf")
```

ファイルを開き model space を取得

"POLYLINE" のエンティティだけ抽出

座標値 (x, y, z) を取得

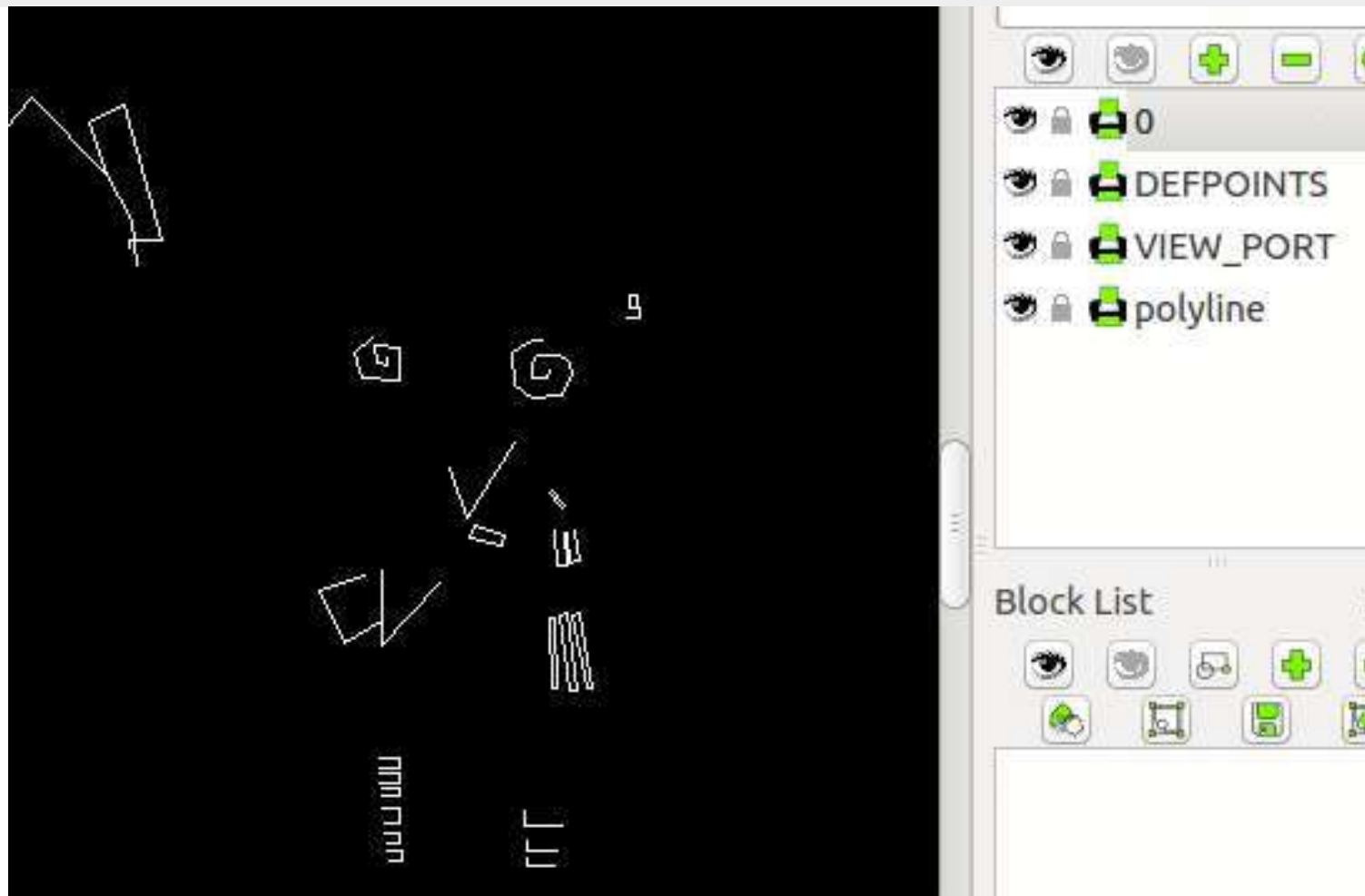
書き出し DXF を開く

書式を設定

値を書き込み・保存

:2 としてるのは
高さ情報除去のため

抽出結果の確認



polyline.dxf を LibreCAD で開いて上記のように表示されたら成功

ezdxf の仕様

- 寸法線のエンティティの値の取得は未実装
- AC1014 (R14) ファイルには未対応

エンティティで異なる座標の取得方法

微妙にメソッド名など座標値の取得方法が異なる。

```
import ezdxf
FR = "hoge.dxf"
msp = ezdxf.readfile(FR).modelspace()

for entity in msp:
    dtype = entity.dxftype()
    if dtype == "POLYLINE":
        list(entity.points())
    if dtype == "LWPOLYLINE":
        list(entity.get_points())
    if dtype == "LINE":
        (line.dxf.start, line.dxf.end)
```


まとめ

- DXF のライブラリを調査
- 自由に使える CAD データを調査
- Python ezdxf ライブラリを紹介
 - インストール
 - DXF ファイルの読み書き手順

質疑応答・コメント（2015-03-16 追記）

- CAD の 2 次元図で面積を求めるときに， DXF に変換してメッシュを切る。線がきちんとつながっていないとできない。そのため，線と線とつながっているかどうか判定したい。なにかいい方法がないか？
 - ▶ 幾何操作のためのライブラリがある。
 - Shapely （ <https://github.com/Toblerity/Shapely> ）
 - Python のたぶん唯一の幾何操作のライブラリ。
 - 要素同士の重なり判定，領域の面積の計算などが簡単にできる。