

# KeTCindy/KeTCindyJSの利用

高遠節夫（東邦大）

明治大教材作成セミナー

2019.09.03

# QR コード



CindyScript

## ヘルプ

- Cinderella のトップメニューで  
ヘルプ > マニュアルを読む
- CindyScript を選ぶ

## 幾何点とリスト点（1）

- 幾何点は，画面で「点を加える」で作る点  
マウスで動かすことができる  
いろいろな属性をもつ

`A.xy`(座標), `A.x` (x 座標), `A.y` (y 座標)

`A.size` (点の大きさ), `A.color` (色 RGB)

「インスペクタ」で設定可能

- 消すには，画面の消しゴムを用いる  
`CindyScript` では消せない

## 幾何点とリスト点（2）

- K<sub>ET</sub>Cindy は幾何点を追加するコマンドをもつ

Putpoint("B",[1,2],B.xy);

Putoncurve("B","sg1");

Slider("S",[0,-1],[5,-1]);

- リスト点は，長さが2のリストで定義される点

pA=[2,3];    pA\_1 (x 座標), pA\_2 (y 座標)

- リスト点は CindyScript で簡単に消せる
- 動きは CindyScript で制御

## 幾何点とリスト点（3）

- リスト点を画面に表示するには  
`Pointdata("1",pt,["Size=3","Color=red"]);`
- 幾何点は  $\text{T}_\text{E}\text{X}$  に出力されない。
- リスト点は `Pointdata` で  $\text{T}_\text{E}\text{X}$  に出力される
- 幾何点は大文字から，リスト点是小文字から始める
- その他の幾何要素（線分，円，2次曲線など）は，`KETCindy` では余り使わない

## リスト (1)

- $[1, 5, 3], 1..5 = [1, 2, 3, 4, 5]$
- リストの長さ `length(list)`  
注意) `1..length(list)` は CindyJS でエラー
- n 番目の要素 `list_n`
- リストの追加  
`list2=append(list1, "a");` (前方追加は `prepend`)
- リストの結合  
`list3=concat(list1, list2);`



## 条件分岐

```
if(条件,  
    真のときの実行文;  
    ,  
    偽のときの実行文;  
);
```

- 条件

`islist(a)` (リストか) , `isstring(a)` (文字列か)  
`contains(list,a)` (list が a を含むか)

## 繰り返し

```
forall(1..10,  
    println(#);  
);  
forall([2,5,4], k,  
    println(k);  
);
```

- 他の繰り返しとして repeat, while  
while は無限ループになりやすく止められない

## リスト (2)

- 要素の削除 `remove(リスト, 要素)`
- 逆順リスト `reverse(リスト)`
- ソーティング `sort(リスト)`
- 式の適用 `apply`  
`list2=apply(list1, 2*#);`
- 要素の選択 `select`  
`list2=select(list1, #<10);`

## 文字列処理（1）

- 連結    "abc" + "de"
- 長さ    length(文字列)
- 部分文字列    substring(文字列, 始め-1, 終り)
- 文字列から数（リスト）へ    parse(文字列)
- 数（リスト）から文字列へ  
    text(数), format(数, 桁数)  
    Textformat(数リスト, 桁数), Sprintf(数, 桁数)

## 文字列処理（2）

- 検索 `indexof(str1,str2)`  
KeTCindy に追加 `Indexall(str1,str2)`
- 分解 `tokenize(str,expr)`  
KeTCindy に追加 `Strsplit(str,expr)`  
`tokenize("a,b,c",","); => ["a","b","c"];`
- 置換 `replace(str,str1,str2);`

## 関数

- 関数定義（例）

```
Dotprod(v1,v2):=(  
    regional(nn, out,tmp);  
    nn=length(v1);  
    tmp=apply(1..nn,v1_#*v2_#);  
    out=sum(tmp);  
    out;  
);
```

- regional にない変数はグローバル

KE TCindy

## ヘルプ

- `setwork.bat(.command)` を実行する  
    KeTCindyGuideJ.pdf (概要)  
    KeTCindyReferenceJ.pdf (関数一覧)
- CindyScript で  
    `Help("関数名");` => ヘルプが表示される
- `KeTCindy` の関数は大文字で始まる。
- **Samples of KeTCindy** を参照



## 手順

- KeTCindy のファイルの 1 つをダブルクリック
- CindyScript を開き，コマンドを記述
- ギヤマークを押す
- 画面に戻り，Figure ボタンを押す  
R,  $\text{T}_{\text{E}}\text{X}$ , pdf ビューアが順に起動する
- 図ファイル (.tex) は fig フォルダにできるので，使いたいフォルダに移動する

## Script の基本形 (1)

- 通常は, draw/figures に記述  
initialization/ketlib はライブラリ読み込み
- ```
Ketinit();//カレントパスはDircdy/fig  
p1=[1,2]; p2=[3,1];  
Listplot("1",[p1,p2]); // sg1 に線分データ  
Plotdata("1","x^2","x"); //gr1 にグラフデータ  
Windispg();
```

## 標準設定の変更

- `Ketinit("");`//カレントパスを Dircdy にする  
`Setfiles(Cdname()+"2");`// ファイル名を変更  
`Usegraphics("pict2e");`//デフォルトは Tpic  
`Addpackage(["emath"]);`//Package を追加  
`Addax(0);`//座標軸を描かない  
`Setax(["l","x","e","y","n","0","sw","Size=1"]);`

## figure スロットと ketlib スロット

- draw にある figure は画面の変化がある度に実行される。
- initialization の ketlib は最初またはギヤマークを押したときだけ。
- 初期化や時間がかかる処理は ketlib においた方がいい

KETCindyJS

## 手順

- トップメニューから  
ファイル > CindyJS に書き出す
- 通常は Dircdy/fig に作られるが、他のファイルを同時に開いていたりすると、カレントディレクトリが変わっている可能性があるから、要注意
- KeTJS オンラインバージョンの HTML を作成
- KeTJS オフラインバージョンの HTML を作成

## 設定など

- 設定

```
Setketcindyjs(["Nolabel=all","Color=offwhite"]);
```

```
Setketcindyjs(["Scale=1","Grid=0.5"]);
```

```
Setketcindyjs(["Fig=y","Axe=n"]);
```

- KeTCindy でのみ実行

```
..... // no ketjs (on,off)
```

- KeTCindyJS でのみ実行

```
//..... // only ketjs (on,off)
```

## アニメーション

- Play, Pause, Rev, Stop ボタンを画面におく  
template4ketcindyjs.cdy を用いるとよい  
リネームするかボタンをコピペ
- アニメボタンのパラメータ値を設定, 取得  
`ss=Animationparam(初期値, 秒速, 範囲);`
- 値が `ss` のときに表示される図を記述  
状態関数 `mf(s)` を定義すると見やすい



## 数式処理の利用 (K<sub>E</sub>TCindy)

- Mxfun(名前, 関数, 引数リスト, オプション)  
`Mxfun("1", "diff", ["x^2", "x"], [""]);`
- Mxtex(名前, Maxima 数式, オプション)  
L<sup>A</sup>T<sub>E</sub>X 数式を出力
- CalcbyM(名前, コマンドリスト, オプション)

## 数式処理の利用 (KETCindyJS)

- KETCindyJS から直接数式処理を呼べない
- KETCindy の計算結果を Ketcindyjsdata で html に書き出すようにしておく

```
Mxfun("1","diff",["sin(x)","x"]);//no ketjs on  
Ketcindyjsdata(["mx1",mx1]);//no ketjs off  
Plotdata("1",mx1,"x");
```

html の csinit に `mx1="cos(x)"` が書き込まれる