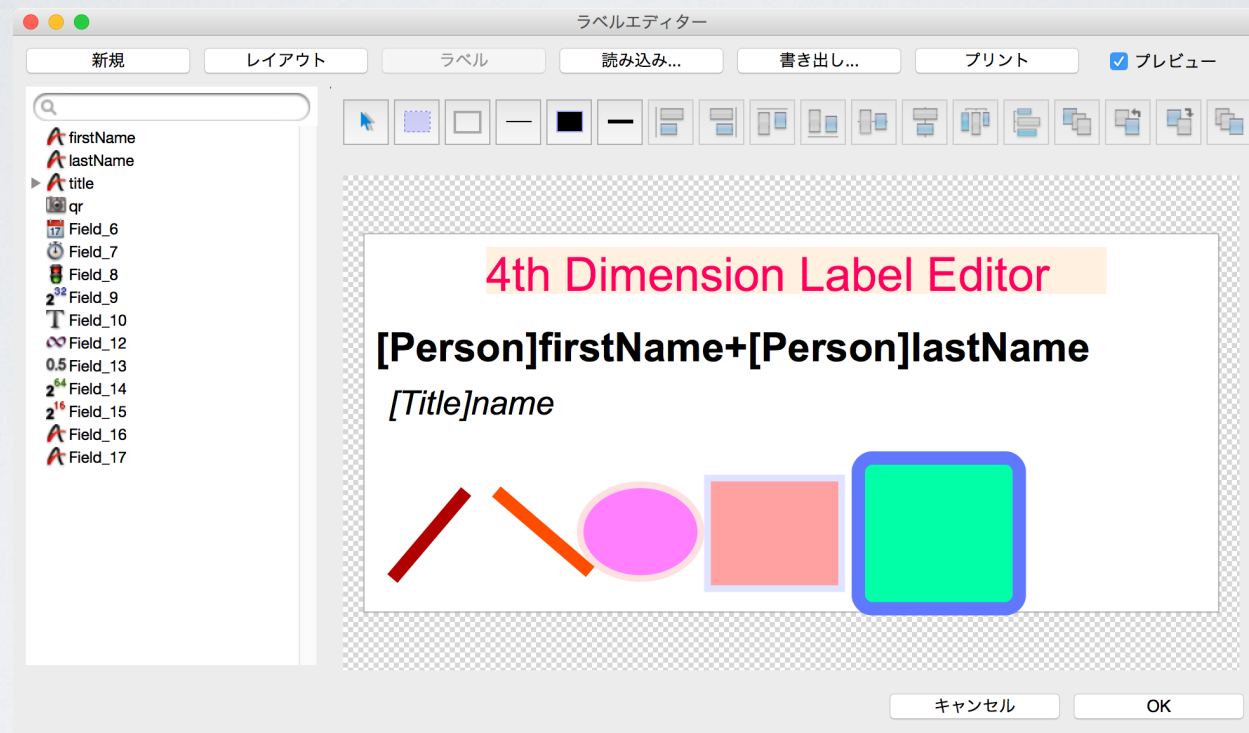


標準SVGコマンドだけで実現!

ラベルエディター

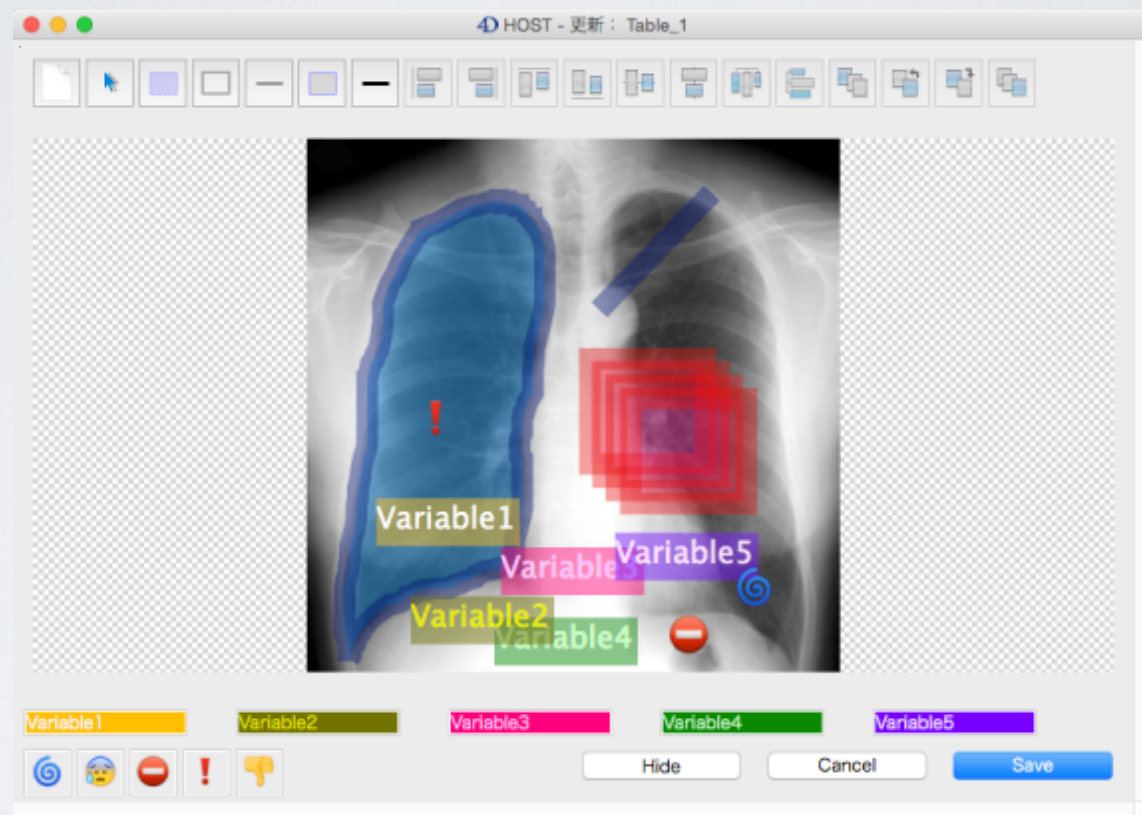
デモンストレーション

ラベルエディター



デモンストレーション

アノテーションエディター



仕様書

ラベル定義ファイル 4LB/IFF

Interchange File Format (IFF) は、汎用ファイルフォーマットの一種。1985年にエレクトロニック・アーツが、コモドールの Amiga 向けに異なるアプリケーション間でのデータ転送を容易にするために開発した。

http://ja.wikipedia.org/wiki/Interchange_File_Format

- バイナリ形式の構造ファイル
- WAV, AVI, AIFF, PNGの元になったフォーマット

仕様書

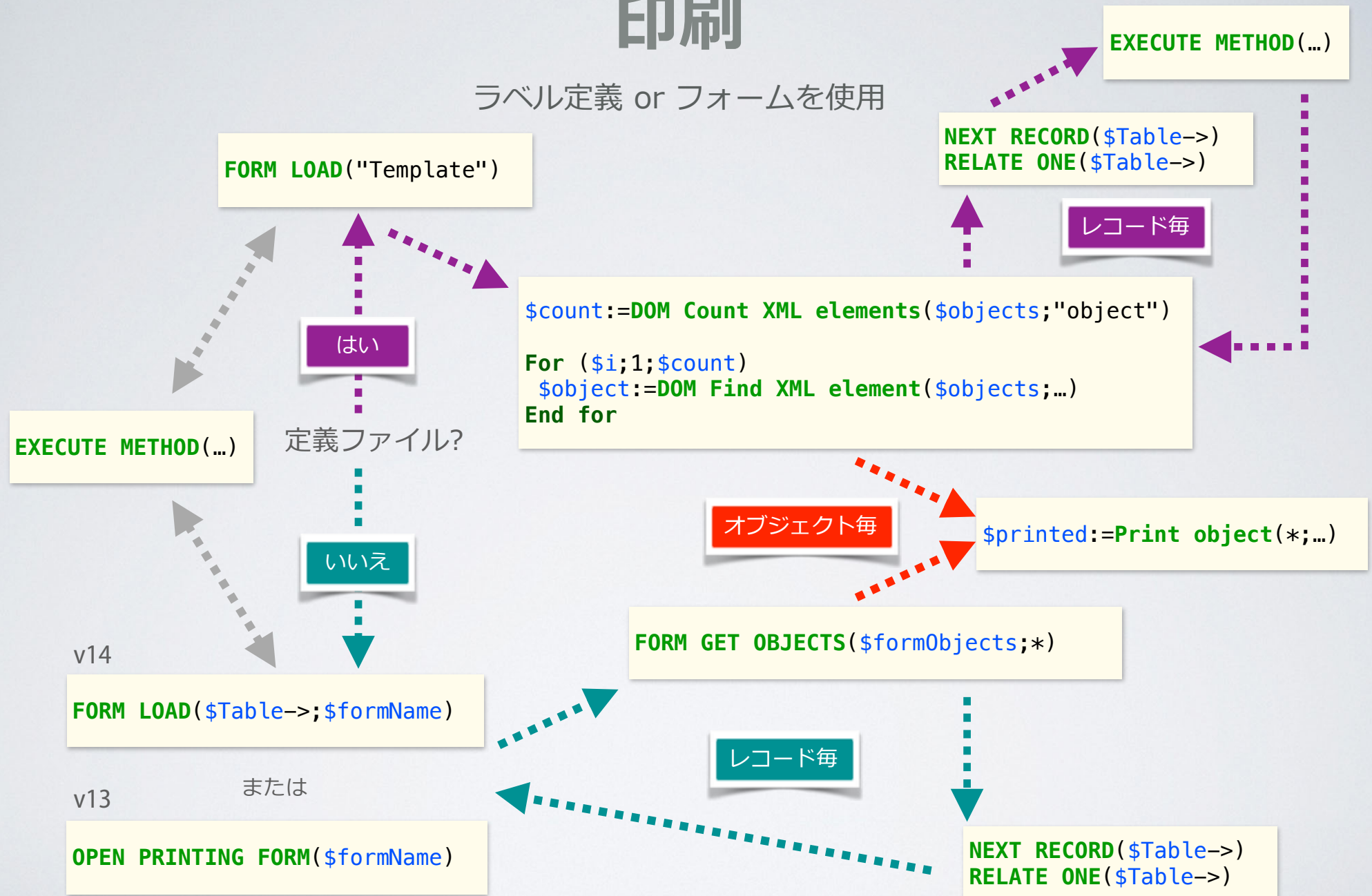
ラベル定義ファイル 4LB/IFF

```
$path:=Get 4D folder(Current resources folder)+"My Label.4LB"  
DOCUMENT TO BLOB($path;$data)  
$dom:=LABEL_Parse_data (->$data)
```

XMLツリーに展開

印刷

ラベル定義 or フォームを使用



おまけTips

SVGピクチャの更新

- SVG SET ATTRIBUTE

- ✧ 画面上にレンダリングされた画像でなければ使用できない
- ✧ バックエンドのXMLを更新しない
- ✧ **高速**

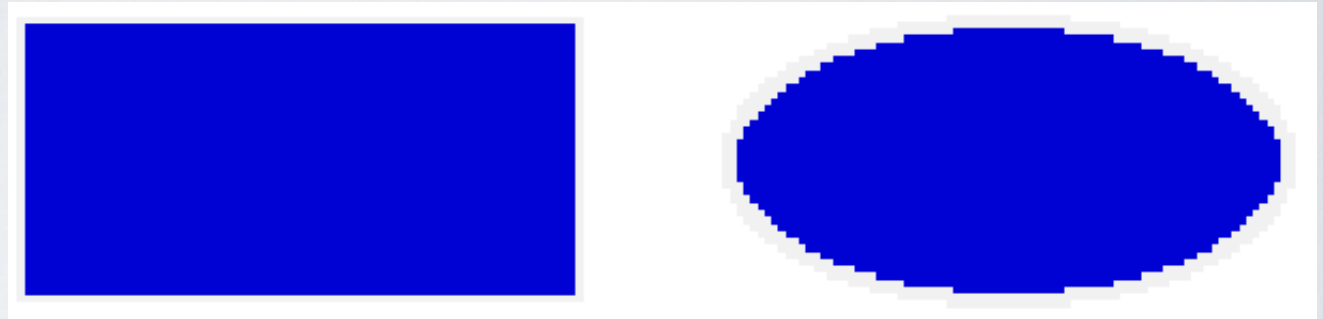
- SVG SET ATTRIBUTE *

- DOM SET XML ATTRIBUTE+SVG EXPORT TO PICTURE

- ✧ オフスクリーンで利用できる
- ✧ バックエンドのXMLを更新する

おまけTips

shape-rendering属性



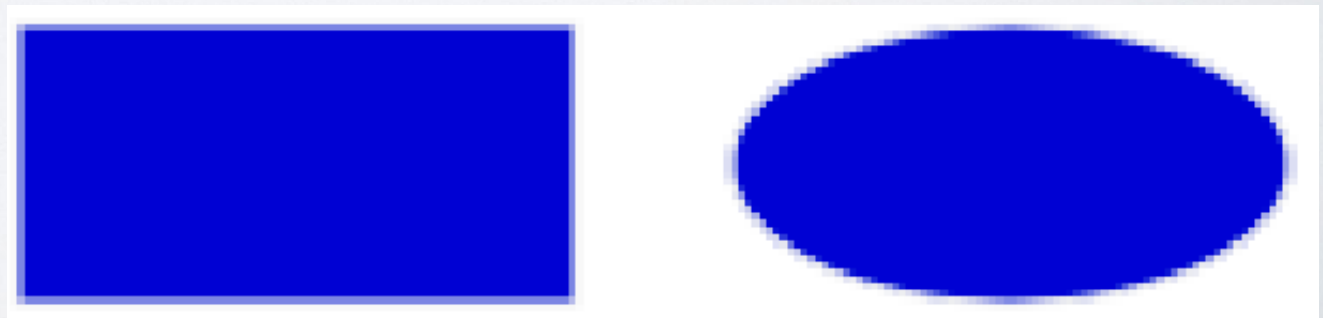
crispEdges

ユーザーエージェントは、レンダリング速度や幾何学的正確性よりも、図表のエッジにあるコントラストを強調するべきであるという指示子。エッジをはっきりさせるため、ユーザーエージェントは、水平または垂直に近い直線、さらには曲線の**アンチエイリアス**を無効にしても良い。ユーザーエージェントは、デバイスのピクセルに合わせて**座標位置**や**線の幅**を揃えることもできる。

geometricPrecision

optimizeSpeed

auto



おまけTips

DOMの移動

- move back
- move back-most
- move front
- move front most

DOM Insert XML element

DOM Insert XML element

DOM Insert XML element

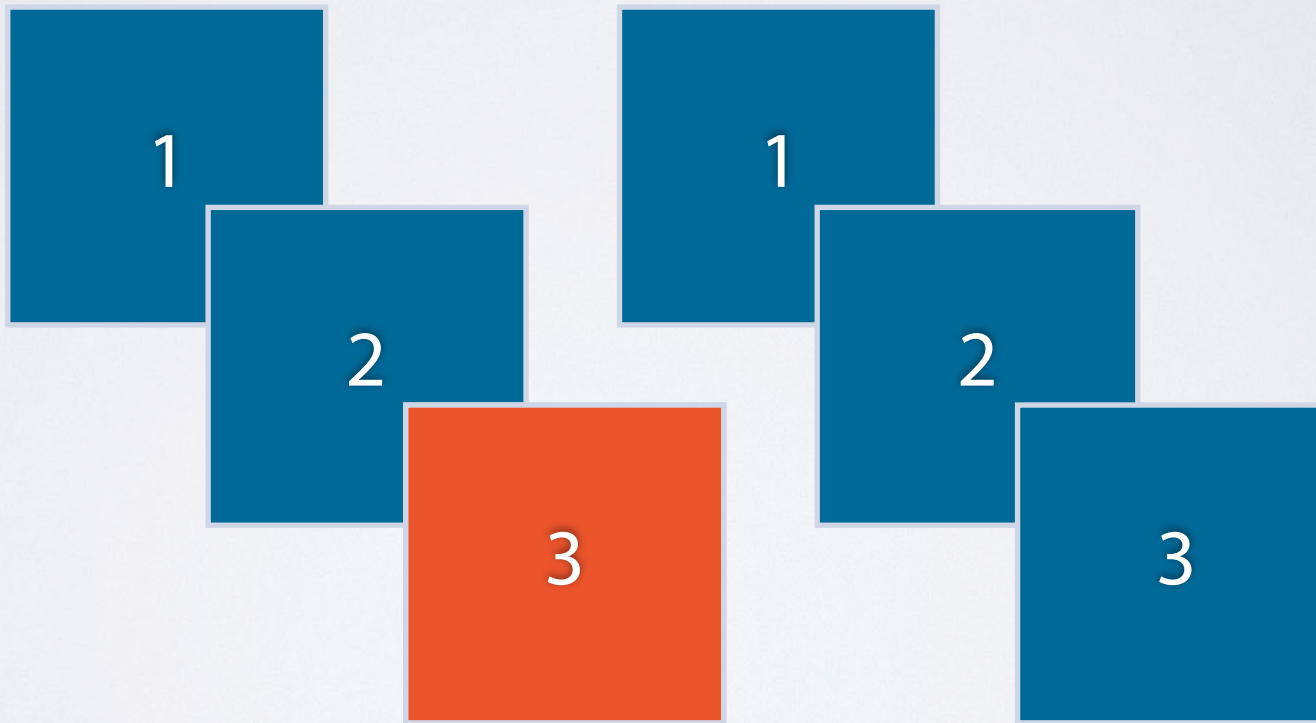
DOM Append XML element

おまけTips

move-back

```
$domParent:=DOM Get parent XML element($dom)  
$newDom:=DOM Insert XML element($domParent;$dom;$level)  
DOM REMOVE XML ELEMENT($dom)
```

2 の手前に 3 を挿入, それから 3 を削除

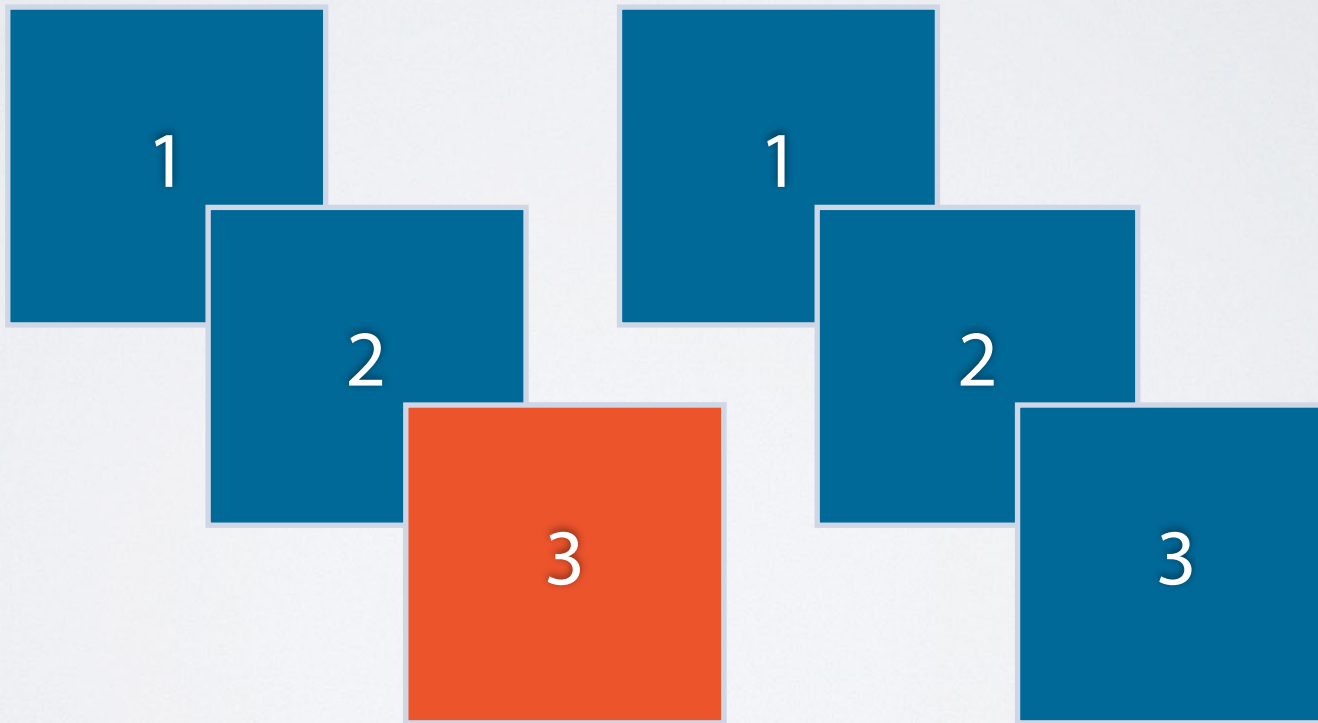


おまけTips

move-back-most

```
$domParent:=DOM Get parent XML element($dom)  
$newDom:=DOM Insert XML element($domParent;$dom;$level)  
DOM REMOVE XML ELEMENT($dom)
```

3 の手前に 1 を挿入, それから 3 を削除

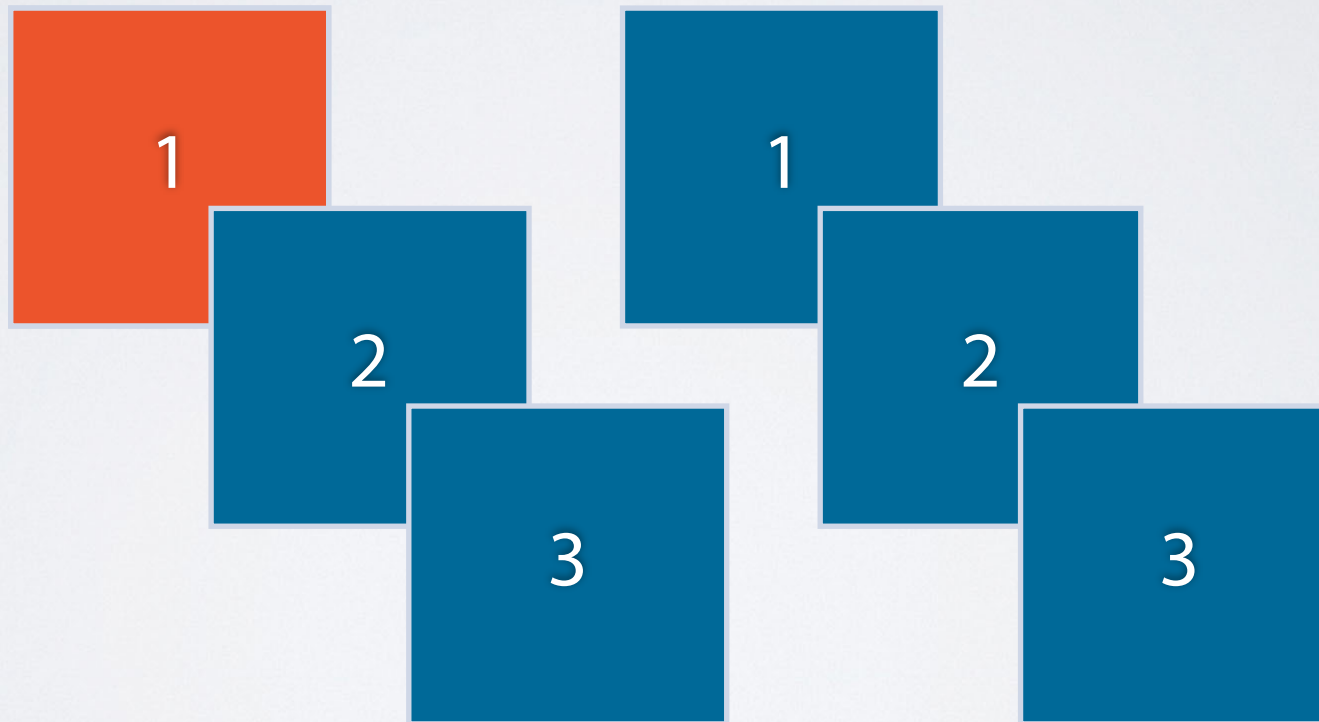


おまけTips

move-front

```
$domParent:=DOM Get parent XML element($dom)  
$newDom:=DOM Insert XML element($domParent;$dom;$level)  
DOM REMOVE XML ELEMENT($dom)
```

3 の手前に 1 を挿入, それから 1 を削除

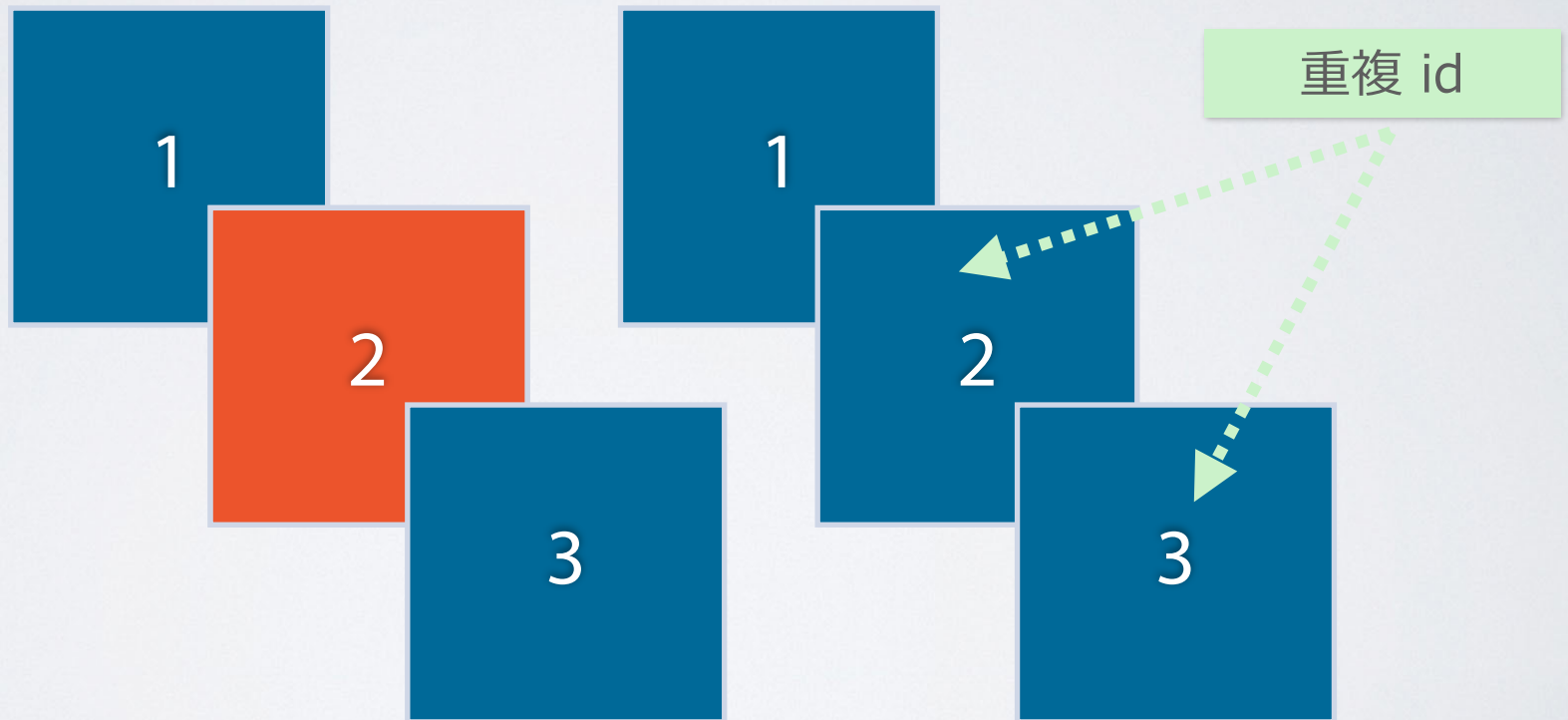


おまけTips

move-front-most

```
$domParent:=DOM Get parent XML element($dom)  
$newDom:=DOM Append XML element($domParent;$dom)  
DOM REMOVE XML ELEMENT($dom)
```

3 の後ろに 2 を追加, それから 2 を削除



おまけTips

DOMの移動

- 挿入/追加により, idの重複が発生する
- DOMを再インデックスしなければならない
- そうでないとDOM Find XML element by IDは混乱した動作になる
- 簡単なのはソースツリーを再構築（再解析）すること

おまけTips

DOMの再インデックス

```
C_TEXT($1)
$dom:=$1


$old:=Method called on error
ON ERR CALL("LABEL_ON_ERROR_CALL")
DOM GET XML ATTRIBUTE BY NAME($dom;"id";$id)
ON ERR CALL($old)

If (OK=1)
  DOM SET XML ATTRIBUTE($dom;"temp-id";$id)
  DOM REMOVE XML ATTRIBUTE($dom;"id")
End if

ARRAY LONGINT($types;0)
ARRAY TEXT($nodes;0)
DOM GET XML CHILD NODES($dom;$types;$nodes)

C_LONGINT($p)
$P:=1
$P:=Find in array($types;XML_ELEMENT;$p)

While ($p#-1)
  //recursive call
  DOM_ELEMENT_CLEAR_ID ($nodes{$p})
  $P:=Find in array($types;XML_ELEMENT;$p+1)
End while
```



remove

```
C_TEXT($1)
$dom:=$1

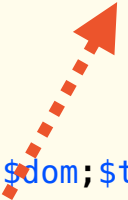
$old:=Method called on error
ON ERR CALL("LABEL_ON_ERROR_CALL")
DOM GET XML ATTRIBUTE BY NAME($dom;"temp-id";$id)
ON ERR CALL($old)

If (OK=1)
  DOM REMOVE XML ATTRIBUTE($dom;"temp-id")
  DOM SET XML ATTRIBUTE($dom;"id";$id)
End if

ARRAY LONGINT($types;0)
ARRAY TEXT($nodes;0)
DOM GET XML CHILD NODES($dom;$types;$nodes)

C_LONGINT($p)
$P:=1
$P:=Find in array($types;XML_ELEMENT;$p)

While ($p#-1)
  //recursive call
  DOM_ELEMENT_RESTORE_ID ($nodes{$p})
  $P:=Find in array($types;XML_ELEMENT;$p+1)
End while
```



restore