

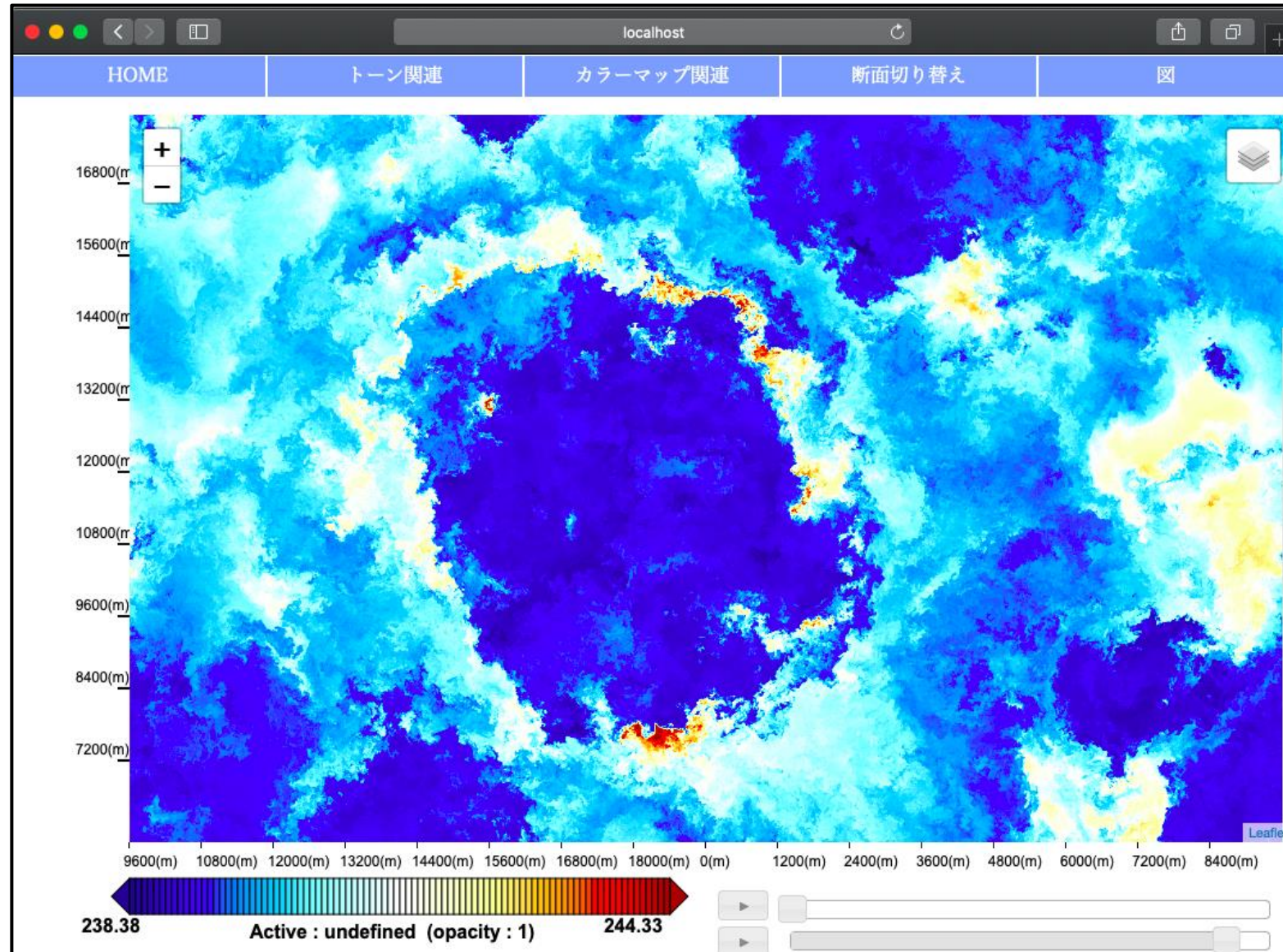
2020 3/13 dcwmtチュートリアル

松江高専 電子情報システム工学専攻 松村和樹

# 本チュートリアルで扱うツールの概要

## <ツールの特徴>

1. 図のシームレスな拡大縮小,  
スクロール
2. DCLの主要機能
  - カラーマップ変更
  - コンター図の描画
  - 簡易アニメーションなど



# 動作環境

- OS
  - UNIX コマンドが利用できる環境を用意してください。
    - Linux, MacOS など
- Web サーバ
  - dcwmt は Web サーバ上で動作します (URL が file:/// から始まっている場合は動作しません).
    - Apache2 など
- Ruby
  - 数値シミュレーションデータ (netCDF 形式) から dcwmt の数値データタイトルを作成するために必要です。
    - RubyGems で gphys パッケージをインストールしてください  
<http://ruby.gfd-dennou.org/products/rubygems/index.htm>

## 使用するnetCDF及びツールのソースをコピー

<https://www.gfd-dennou.org/arch/dcwmt> より, tar.gz を取得.

```
$ wget http://www.gfd-dennou.org/arch/dcwmt/src/dcwmt_current.tar.gz
$ tar dcwmt_current.tar.gz
$ cd dcwmt_20200312
```

サンプル用 netCDF ファイルの取得 (パスワードは口頭で連絡) (1行で書く)

```
$ wget --http-user=XXXX --http-password=XXXX
http://www.gfd-dennou.org/arch/dcwmt/netcdf/h06000m_0.2_dx5_t14.5.nc
```

[参考: netCDF ファイルが取得できない場合] 作成済みタイルをwgetする. 22 MB.

```
$ wget http://www.gfd-dennou.org/arch/dcwmt/sample/2020-03-
13_matumura/tile.tar.gz
```

# 1枚の図をブラウザに表示する

- h06000m\_0.2\_dx5\_t14.5.ncをブラウザに描画する
  - 特定の高度, 時刻におけるXY断面の値が格納されたファイル
- PT(温位)のデータタイルを生成 (※5分程度かかります)  
\$ ./makeTile.rb h06000m\_0.2\_dx5\_t14.5.nc@PT tile/PT/t=14.5/h=6000m

# makeTile.rb(データタイル生成スクリプト)の使い方

`./makeTile.rb filename@value,dimname=pos1, dimname=pos2 dirname`

- 読み込むnetCDFのファイル名および変数の指定
- 軸が3つ以上含まれるファイルのときは2次元になるように軸の値を固定する

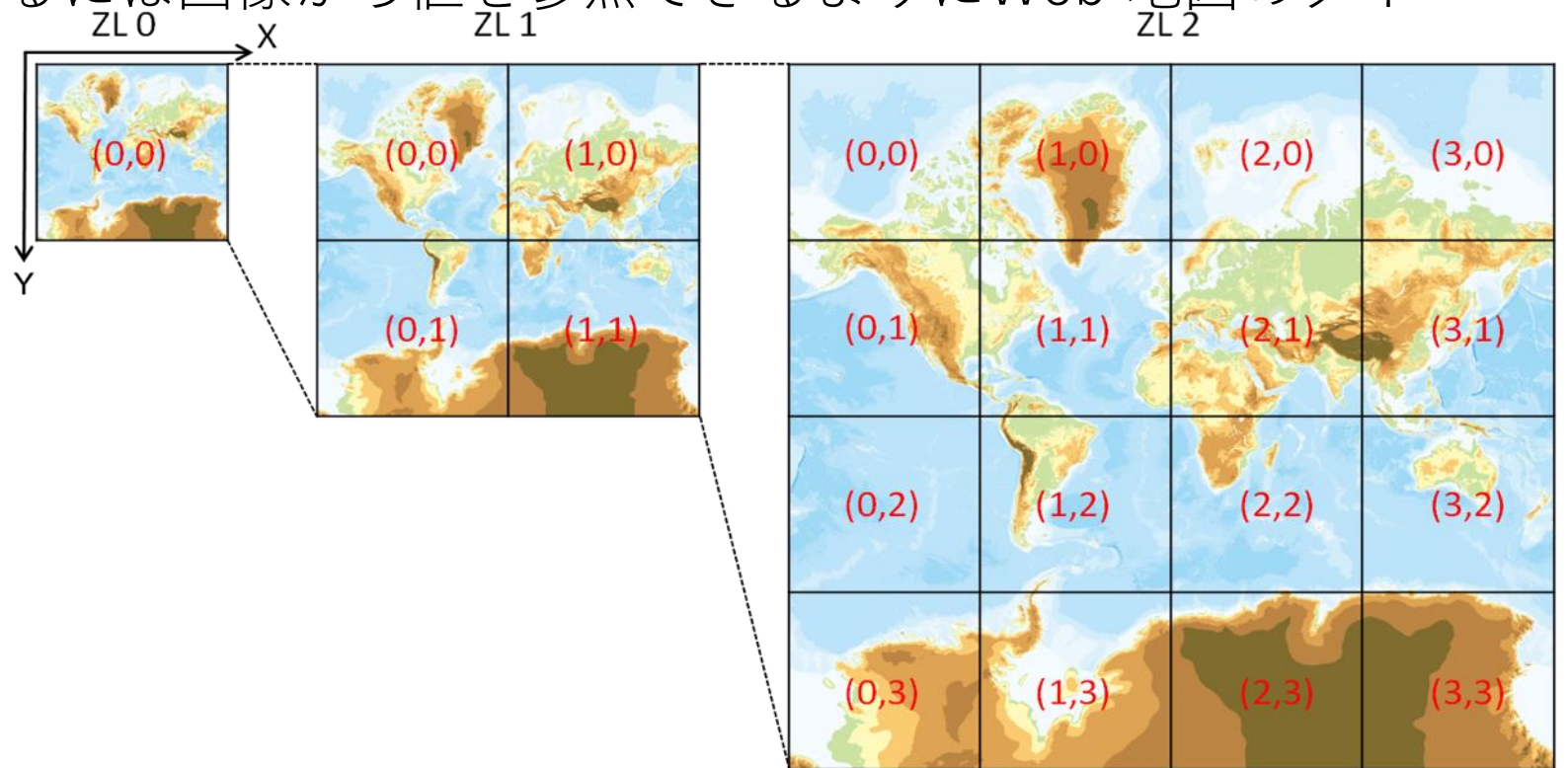
- タイルの格納先ディレクトリ
- 省略時はカレントディレクトリになる

実行例1 `./makeTile.rb sample.nc@V,z=1000,t=0`

実行例2 `./makeTile.rb sample.nc@U,x=1000,t=0 ~/tile`

# Web地図におけるタイル

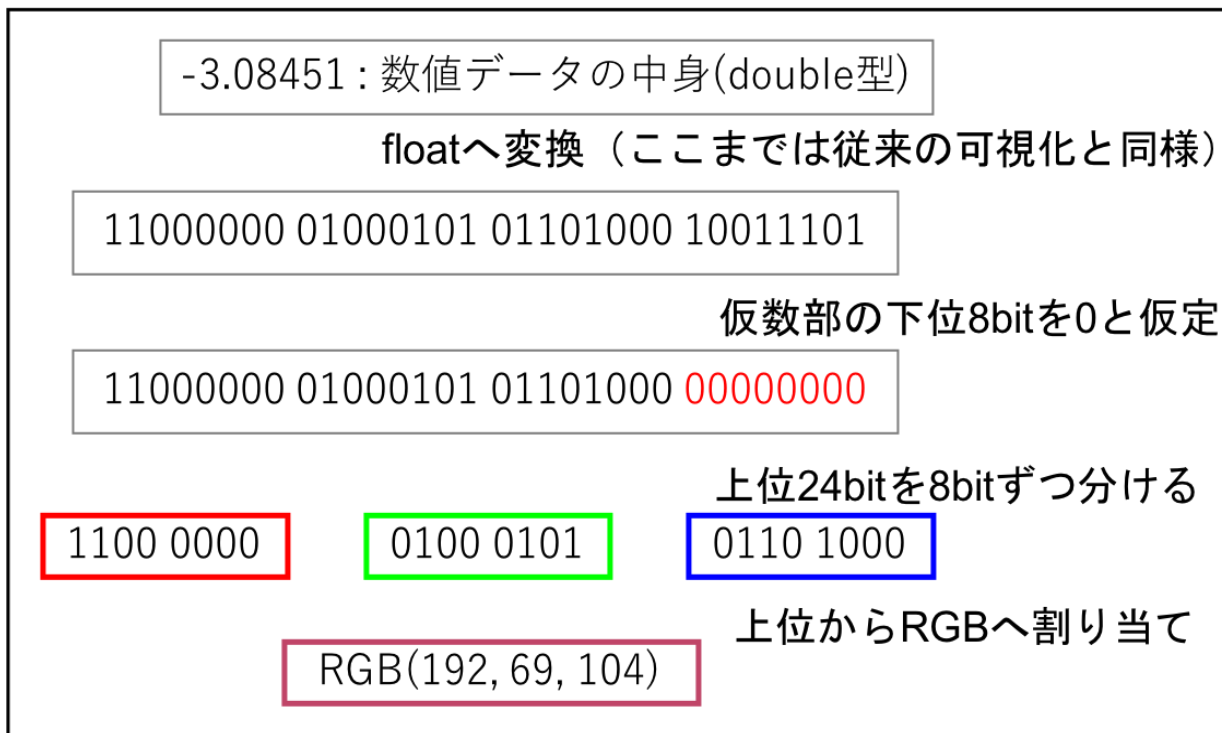
- Web地図で表示される画像は、一般に拡大レベル毎に分割されており、その中から表示領域に入ったものだけを読み込むことでスムーズな操作を実現。
- DCLの主要機能を実現するには画像から値を参照できるようにWeb地図のタイルを拡張する必要がある



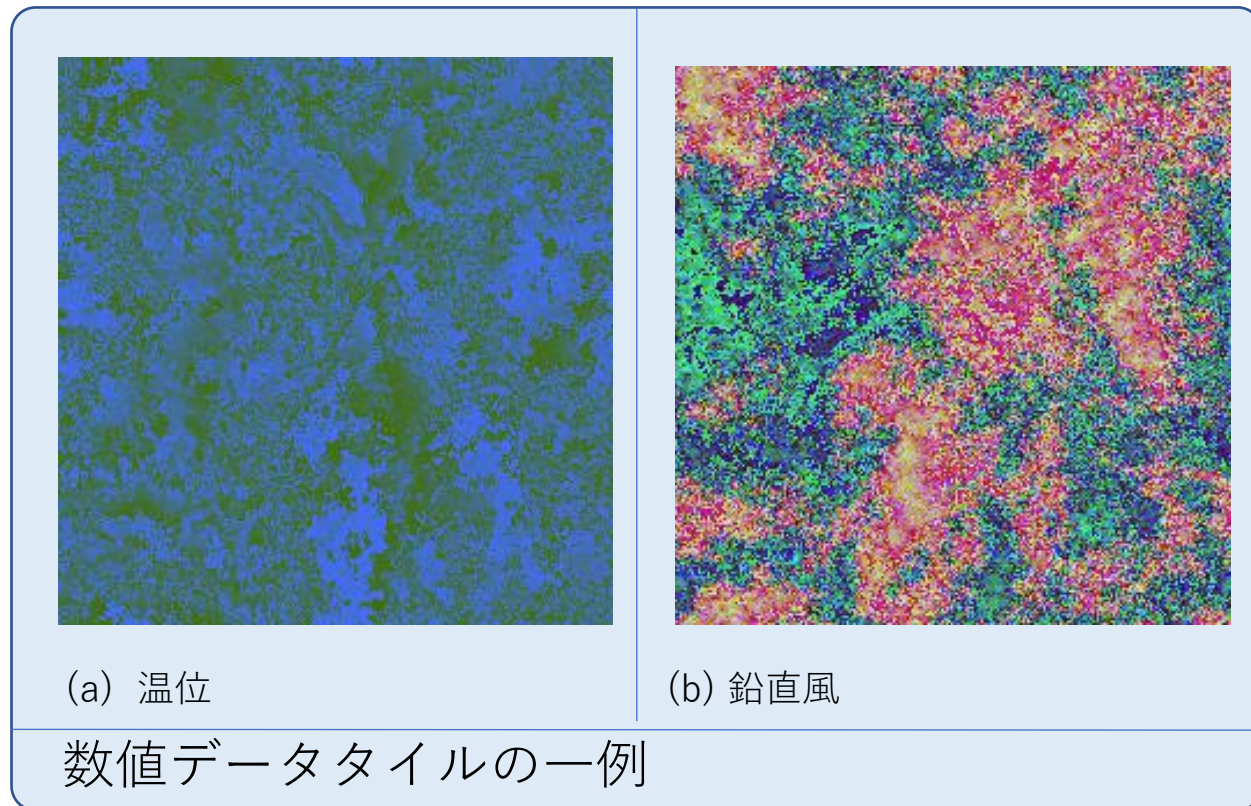
Web地図における拡大度別のタイル

# netCDFのタイル化

- Web地図上で数値を参照可能とするための拡張として**数値データタイル**を定義。
  - 数値データタイルは国土地理院の地理院地図で用いられてる**標高タイル**を拡張した。**標高タイル**では**整数(標高値x100)**と**RGB値**が一対一に対応している。



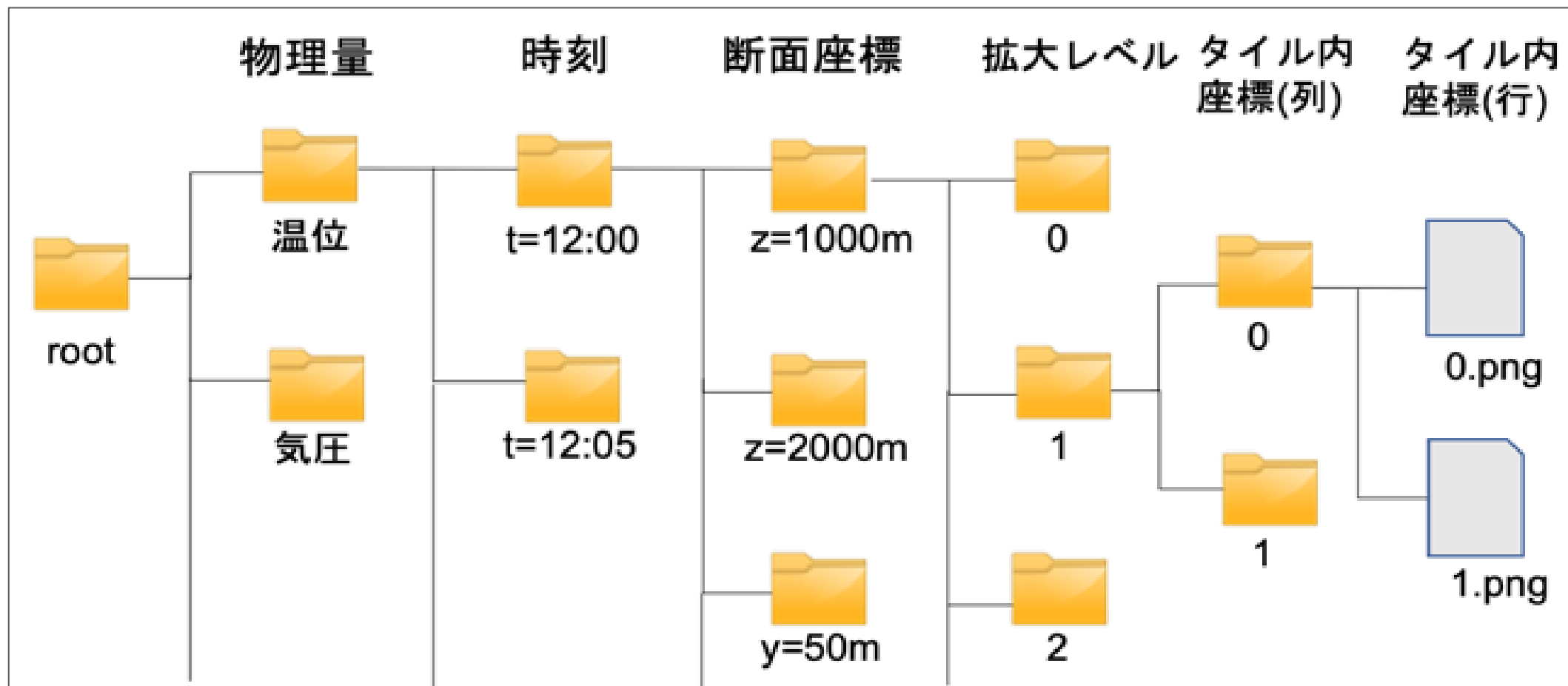
数値データタイルにおける**実数**と**RGB値**の対応





# データタイトルの格納ツリー

- 以下のディレクトリツリーでデータタイトルを保存



数値データタイトルのディレクトリツリー

# 設定ファイル(define.js)の記述

- tool\_src/define.jsに以下を記述 (今回は入力済み)

```
var scale_x=[0,18200]; //x軸の計算領域(0m~18200m)
var scale_y=[0,18200]; //y軸の計算領域(0m~18200m)
var dir_root = "../tile" //タイルが格納されているディレクトリ
var value_name=["PT"]; //物理量ディレクトリ(トーン・コンター図描画用)
var value_name_vec=[]; //物理量ディレクトリ(ベクトル図描画用)
var dir_time=["t=14.5"]; //時間ディレクトリ
var dir_dim=["h=6000m"] //座標軸ディレクトリ
var tile_size_x = 240; //データタイルのヨコ解像度
var tile_size_y = 240; //データタイルのヨコ解像度
var continuous = true; //true:図を縦横に周期的に表示 false: 横のみ周期的に表示
var max_zoom = 4 //何段階拡大させるか
```

## 他の物理量についても表示する場合

- その他のデータタイルを生成

```
$ ./makeTile.rb h06000m_0.2_dx5_t14.5.nc@DENS tile/DENS/t=14.5/h=06000m
```

```
$ ./makeTile.rb h06000m_0.2_dx5_t14.5.nc@U tile/U/t=14.5/h=06000m
```

```
$ ./makeTile.rb h06000m_0.2_dx5_t14.5.nc@V tile/V/t=14.5/h=06000m
```

```
$ ./makeTile.rb h06000m_0.2_dx5_t14.5.nc@W tile/W/t=14.5/h=06000m
```

- tool\_src/define.jsの以下を変更 (他の項目はPTだけの表示と同様)

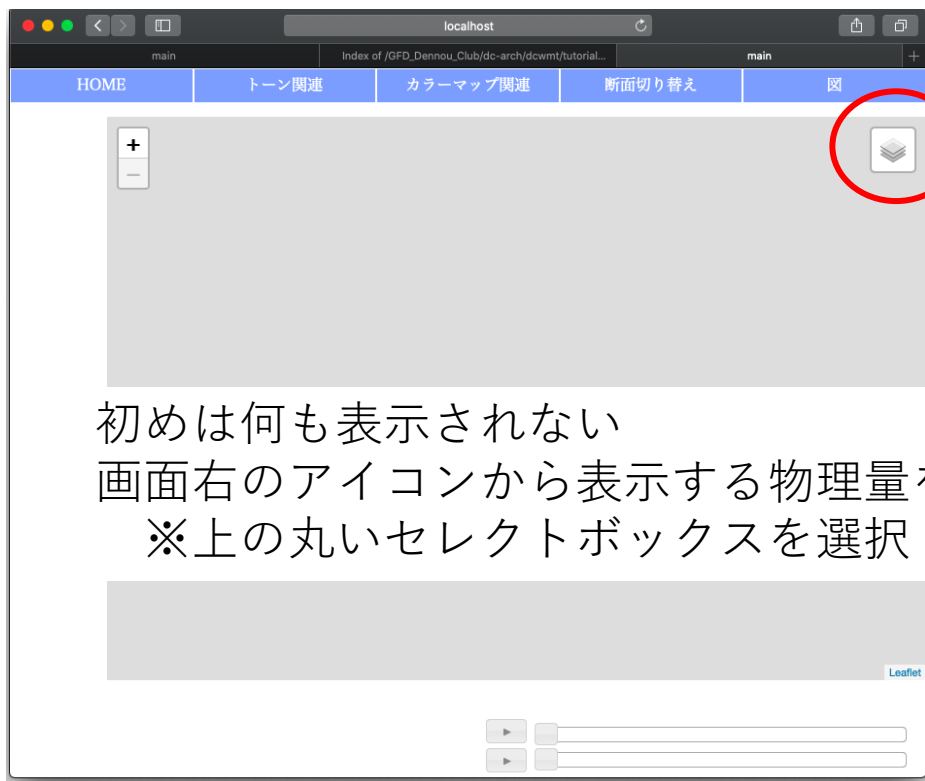
```
var value_name=["PT","DENS","U","V","W"]; //物理量ディレクトリ(トーン・コン  
タ－図描画用)
```

```
var value_name_vec=["U","V"]; //物理量ディレクトリ(ベクトル図描画用)
```

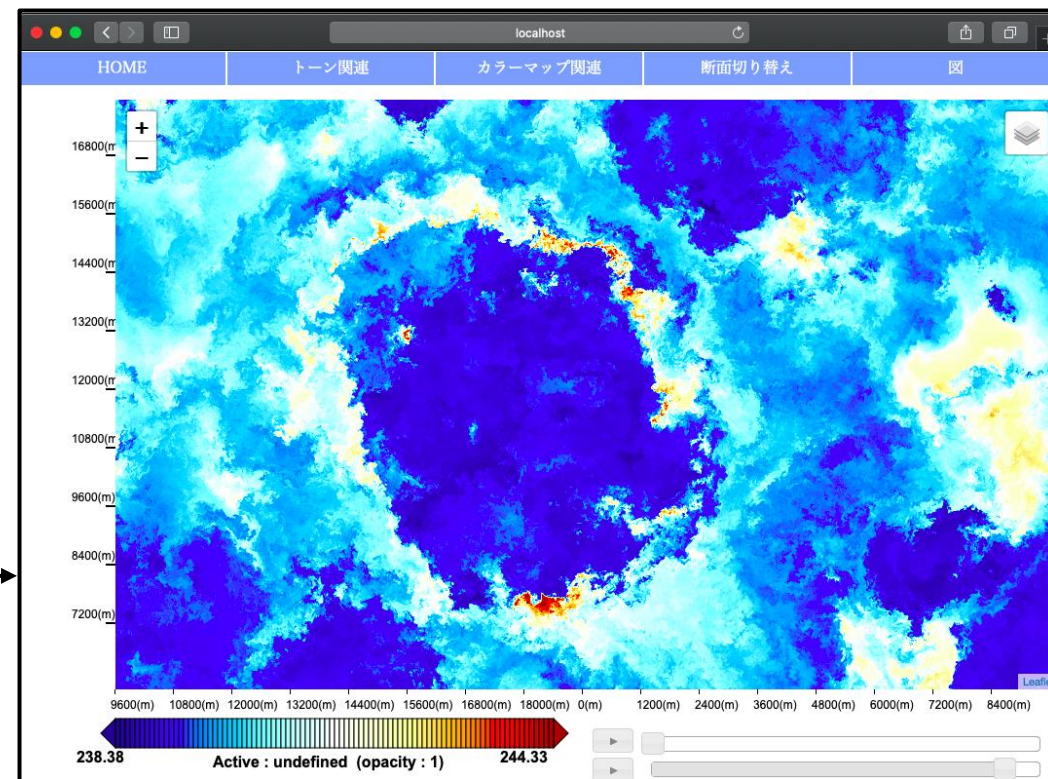
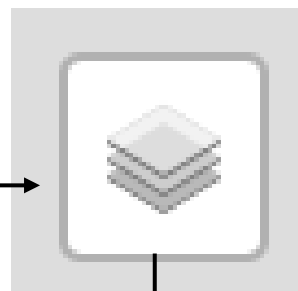
# ブラウザで確認

本ツールを動かすためには、dcwmt の提供する tools フォルダを Webサーバのドキュメントルート以下に置く必要があります。

- ブラウザで<http://myhost/path/tools/main.html>にアクセス ([myhost/path](#) は適宜読み替え)



初めは何も表示されない  
画面右のアイコンから表示する物理量を選択  
※上の丸いセレクトボックスを選択



# 時間・空間の4次元データを表示する

- 使用するデータとして以下を用いる  
dennou-h.gfd-dennou.org:/GFD\_Dennou\_Work12/sugiyama/VENUS/  
Baker\_X128Y128Z20\_dx\_200m\_dz\_125m/TIME\_0\*
- 時間軸, 座標軸の分makeTile.rbをループ実行し, define.js(設定ファイル)を出力する  
**auto.rb**を用意.
  - 実行には数時間かかるため, 本チュートリアルでは事前に用意したタイルと設定ファイルを利用.
  - 作成済みタイルをwgetする (2.5GB).  
\$ wget http://www.gfd-dennou.org/arch/dcwmt/sample/2020-03-13\_matumura/tile\_4dim.tar.gz

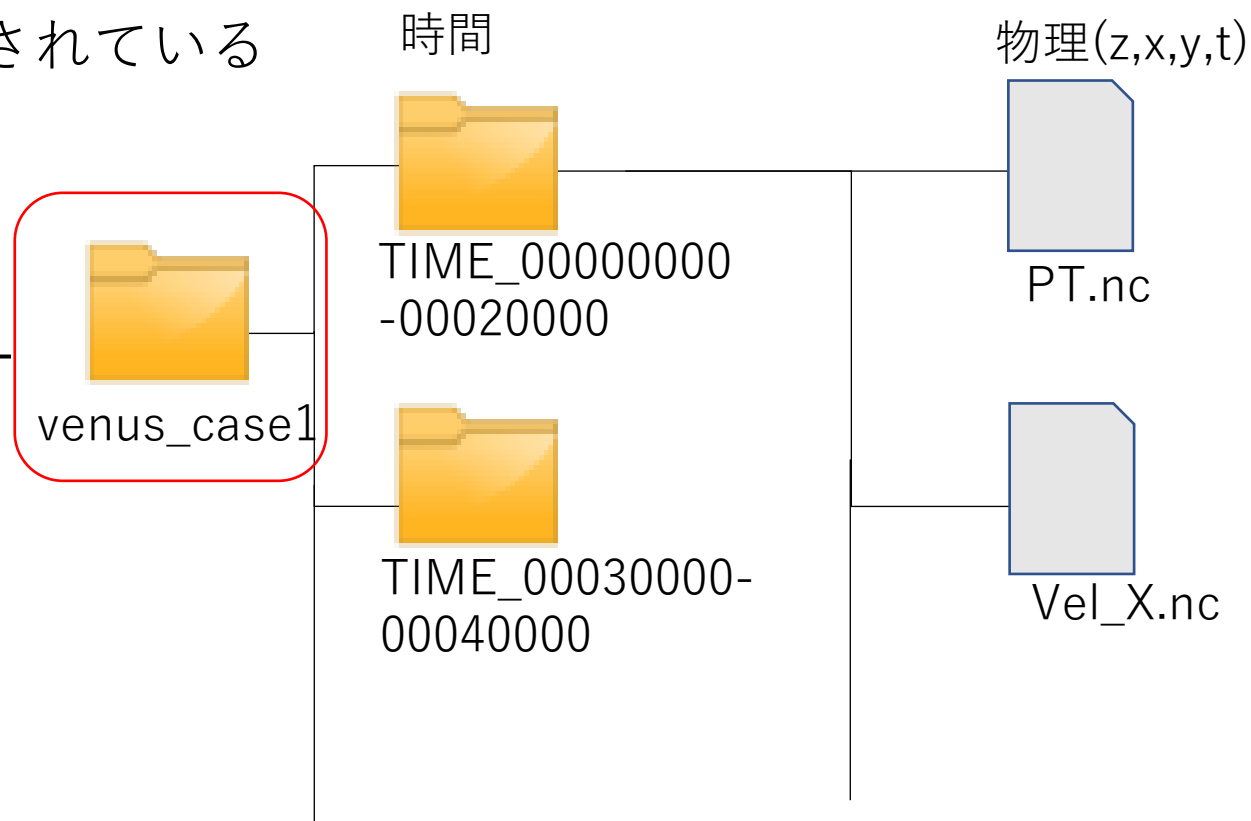
# [参考] auto.rbの使い方

- データに応じてスクリプト内の変数を変更して(後述)実行する.
  - 右のようなツリーでnetCDFが保存されている場合のみ実行可能.

実行例

`./auto.rb venus_case1/`

時間階層が格納されている  
ディレクト名前を指定



## [参考] auto.rbのユーザ定義部分の記述例(XY断面)

```
4   scale_x=[0,128000]           #X軸の計算領域[m] / [最小値,最大値]
5   scale_y=[0,128000]           #Y軸の計算領域[m] / [最小値,最大値]
6   grid_x = 640;                #X軸の格子点数
7   grid_y = 640;                #Y軸の格子点数
8   val_name = ["x","y","z","t"] #図のヨコ,タテ,断面,時間に対応する座標軸の変数名
9   dim=[47200,51000,55000,60000] #断面にするZ軸の座標
10  value = ["PTemp"]            #トーン図かコンター図に使う物理量
11  vec = ["VelX", "VelY"]       #ベクトル図に使う物理量
```

## [参考] auto.rbのユーザ定義部分の記述例(YZ断面)

```
4   scale_x=[0,128000]           #netCDFのY軸の計算領域[m]
5   scale_y=[40000, 60000]       #netCDFのZ軸の計算領域[m]
6   grid_x = 640;                #Y軸の格子点数
7   grid_y = 160;                #Z軸の格子点数
8   val_name = ["y","z","x","t"] #図のヨコ,タテ,断面時間に対応する座標軸の変数名
9   dim=[100,40000,80000,120000] #断面にするX軸の座標
10  value = ["PTemp"]            #トーン図かコンター図に使う物理量
11  vec = ["VelX", "VelY"]       #ベクトル図に使う物理量
```



## 設定ファイル及びタイルを用意する

- 設定ファイルをauto.rbが出力したものに置き換える

```
$ cp -r tools tools2
```

```
$ cd tools2/
```

```
$ mv define.js define_tmp.js
```

```
$ mv define_4dim_XY.js define.js
```

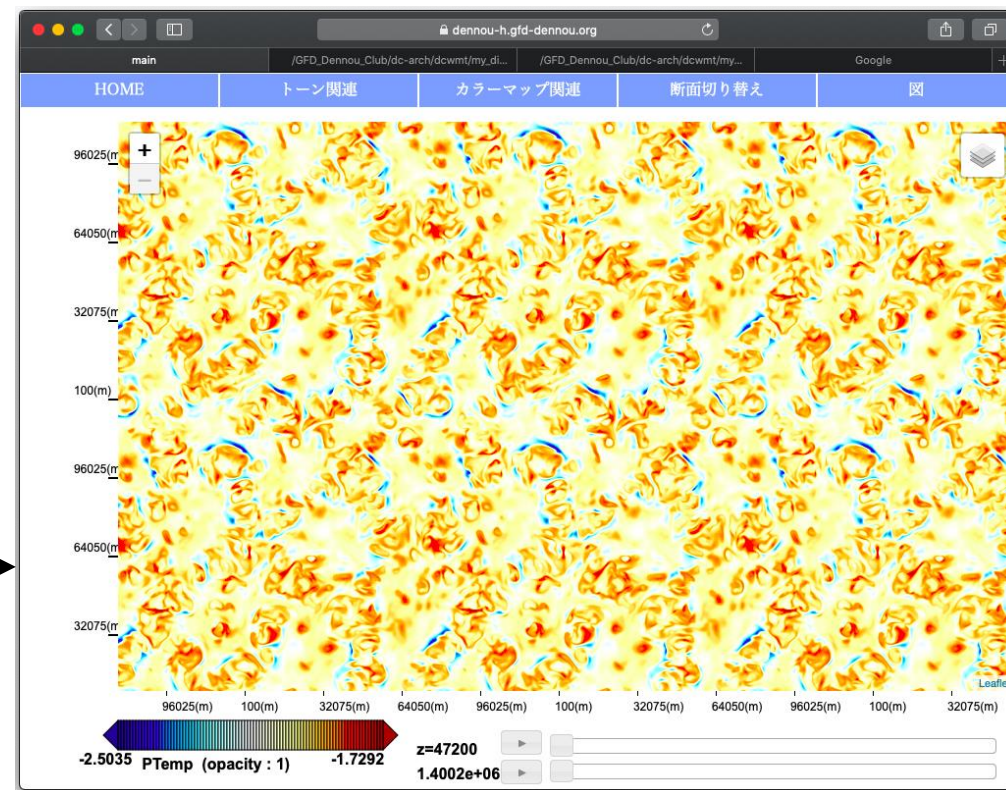
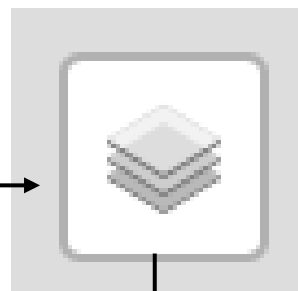
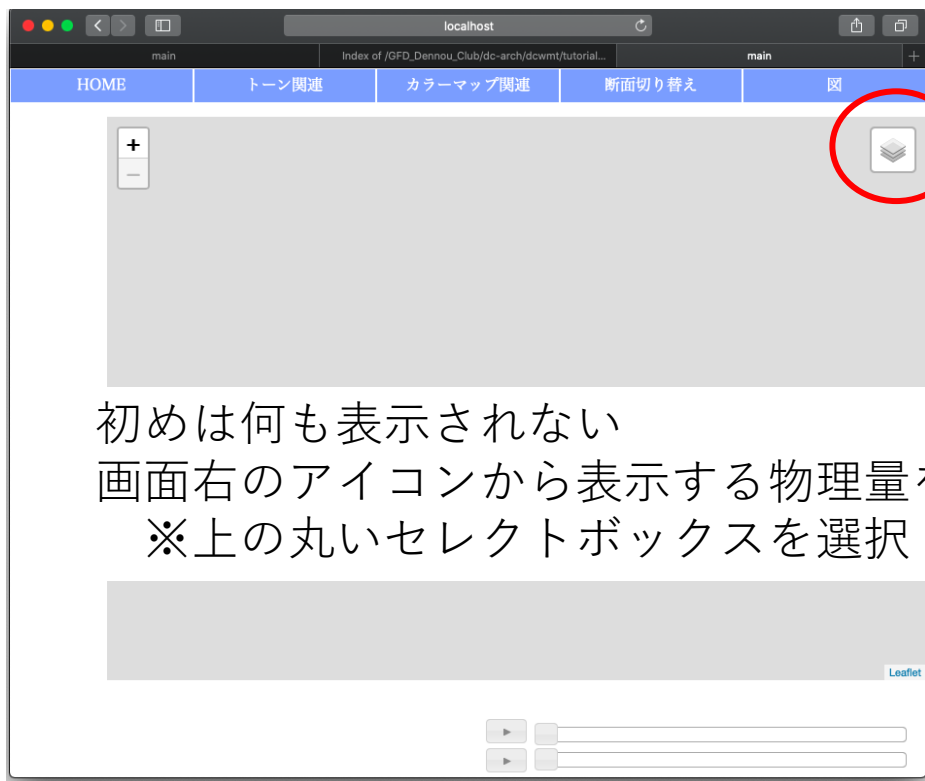
- auto.rbが生成したタイルはtile\_4dim/以下に置かれている

```
$ ls tile_4dim/
```

```
PTemp VelX VelY VelZ
```

# ブラウザで確認(XY断面)

- ブラウザで<http://myhost/path/tools2/main.html>にアクセス ([myhost/path](#) は適宜読み替え)



## XZ断面でも表示する

- htmlファイルを別途用意

```
$ cp -r tools tools3
```

```
$ cd tools3/
```

```
$ mv define_4dim_XZ.js define.js
```

- ブラウザで<http://myhost/path/tools3/main.html>にアクセス ([myhost/path](#) は適宜読み替え)

以下付録

# 設定ファイル(define\_tile.js)の記述

/Library/WebServer/Documents/tutorial/define\_tile.js

```
1行目  var pt = L.gridLayer.numDataGroup('tile/PT',{  
2行目      tileSize : new L.Point(240, 240)  
3行目  });
```

L.gridLayer.numDataGroup : カラートーン図及びコンター図の描画クラス

```
var 変数名(任意) = L.gridLayer.numDataGroup('図の描画に使うタイルの格納場所',{  
      tileSize : new L.Point(タイルの1枚の横の解像度, 縦の解像度)  
});
```

※読み込むタイルはhtml及びJavaScriptのファイルと同一オリジンに保存されている必要がある

# 設定ファイル(define\_tile.js)の記述

/Library/WebServer/Documents/tutorial/define\_tile.js

```
4行目  var uv = L.gridLayer.vectorNumData('tile/U', 'tile/V'){  
5行目     tileSize : new L.Point(240, 240)  
6行目  });
```

L.gridLayer.vectorNumData : カラートーン図及びコンター図の描画クラス

矢印の横

矢印の縦

```
var 変数名(任意) = L.gridLayer.vectorNumData('タイルの格納場所1', 'タイルの格納場所2',{  
    tileSize : new L.Point(タイルの1枚の横の解像度, 縦の解像度)  
});
```

※矢印の描画のためにタイルを2セット指定すること以外はL.gridLayer.numDataGroupと同じ

# 設定ファイル(define\_tile.js)の記述

/Library/WebServer/Documents/tutorial/define\_tile.js

```
7行目  var baseMaps    = {"PT": pt, "U-V": vec_uv};  
8行目  var overlayMaps = {"PT": pt, "U-V": vec_uv};
```

- {}内に”任意の名前”:先に定義した変数名を列挙する(何個でも可)
- ここで指定したタイルをブラウザに表示させる

# Github()からツール本体のソースコードをクローンする

The screenshot shows a web browser window with the URL `github.com/gfd-dennou-club/dcwmt/tree/20_0306tutorial`. The browser tab is titled "GitHub - gfd-dennou-club/dcwmt". The page content includes a branch selector set to "20\_0306tutorial", a "New pull request" button, and a green "Clone or download" button. Below this, there is a commit history table with columns for file names, commit messages, and timestamps.

File Name	Commit Message	Timestamp
L.LayerCtl.js	matsumura first commit	14 minutes ago
L.gridLayer.numData.js	matsumura first commit	14 minutes ago
L.gridLayer.vectorNumData.js	matsumura first commit	14 minutes ago
colormap.js	matsumura first commit	14 minutes ago
define_tile.js	matsumura first commit	14 minutes ago
design.css	matsumura first commit	14 minutes ago
dialog.js	matsumura first commit	14 minutes ago
func.js	matsumura first commit	14 minutes ago
main.html	matsumura first commit	14 minutes ago
main.js	matsumura first commit	14 minutes ago
slider.js	matsumura first commit	14 minutes ago
tmp.html	matsumura first commit	14 minutes ago

1. [https://github.com/gfd-dennou-club/dcwmt/tree/20\\_0306tutorial](https://github.com/gfd-dennou-club/dcwmt/tree/20_0306tutorial)にアクセス
2. branchは20\_0306tutorialを選択
3. Clone or download -> Download Zipよりダウンロード