

Pythonで 3次元CGを作りたい 人のための PyVista 入門


Speaker:

小山哲央

Date:

2024-09-28

自己紹介

-  [@tkoyama010](#)
- 3D 可視化ライブラリ
[@pyvista](#) メンテナ兼ドキュメント翻訳者



もくじ

- 概要
 - PythonでCG作成
- 基礎編
 - モデリング
 - テクスチャ
 - ライティング
 - マテリアル
- 応用編
 - 空間のデータ分析
 - 空間のデータ可視化

PythonでCG作成

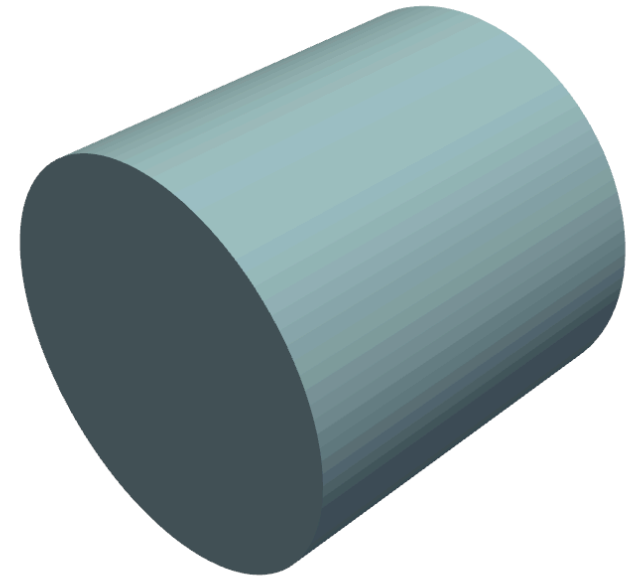
- 概要
 - 🙌 PythonでCG作成
- 基礎編
 - モデリング
 - テクスチャ
 - ライティング
 - マテリアル
- 応用編
 - 空間のデータ分析
 - 空間のデータ可視化

モデリング

- 仮想3次元空間上に個々のポリゴンをつくる。
- CGを作る際に最も基本的な作業。
- 仮想3次元空間上に自分の作成したい物体を配置することができる。

Static Scene

Interactive Scene



テクスチャ

- 質感を表現するための画像をモデルに貼り付ける。
- モデリングで作成したオブジェクトに、テクスチャを貼り付けることで、CGをよりリアルに表現することができる。

Static Scene

Interactive Scene

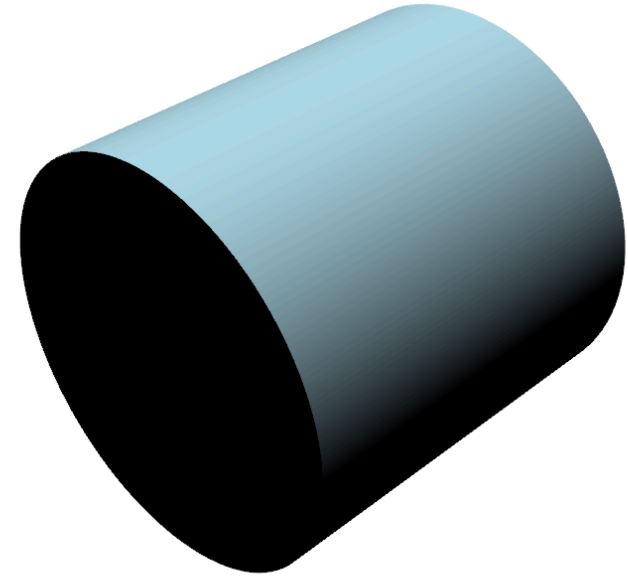


ライティング

- 仮想3D空間に光を配置してオブジェクトを照らす。
- 光源を配置することで、モデリングしたオブジェクトに影をつけることができる。

Static Scene

Interactive Scene



PyVistaとは？

1. Pythonフレンドリな3D可視化ライブラリです。
2. MatplotlibやPandasのAPIに似ています。
3. Matplotlibで実現できないCGの表現もPyVistaで実現できます。
4. Jupyter NotebookやSphinxでのインタラクティブな可視化もサポートしています。

```
$ pip install "pyvista[all]"
```


モデリング

- 概要

- PythonでCG作成

- 基礎編

- 📌 モデリング
- テクスチャ
- ライティング
- マテリアル

- 応用編

- 空間のデータ分析
- 空間のデータ可視化

モデリング

```
# PyVistaをインポートする。
```

```
import pyvista as pv
```

```
# 円柱のモデルを作成する。
```

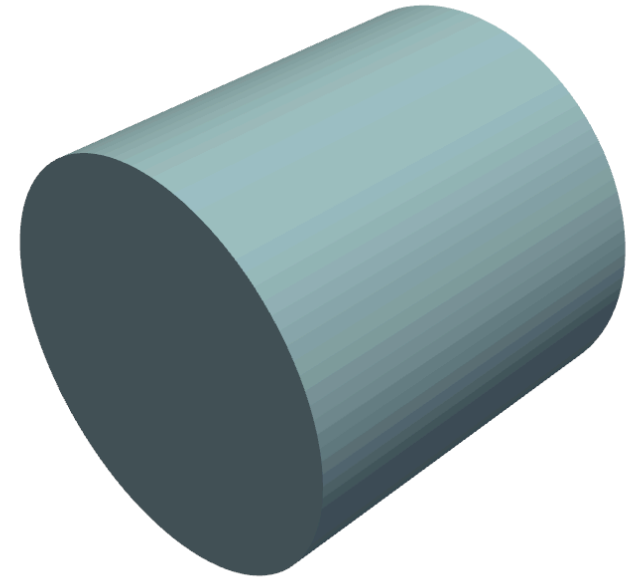
```
mesh = pv.Cylinder()
```

```
# 円柱のモデルを描画する。
```

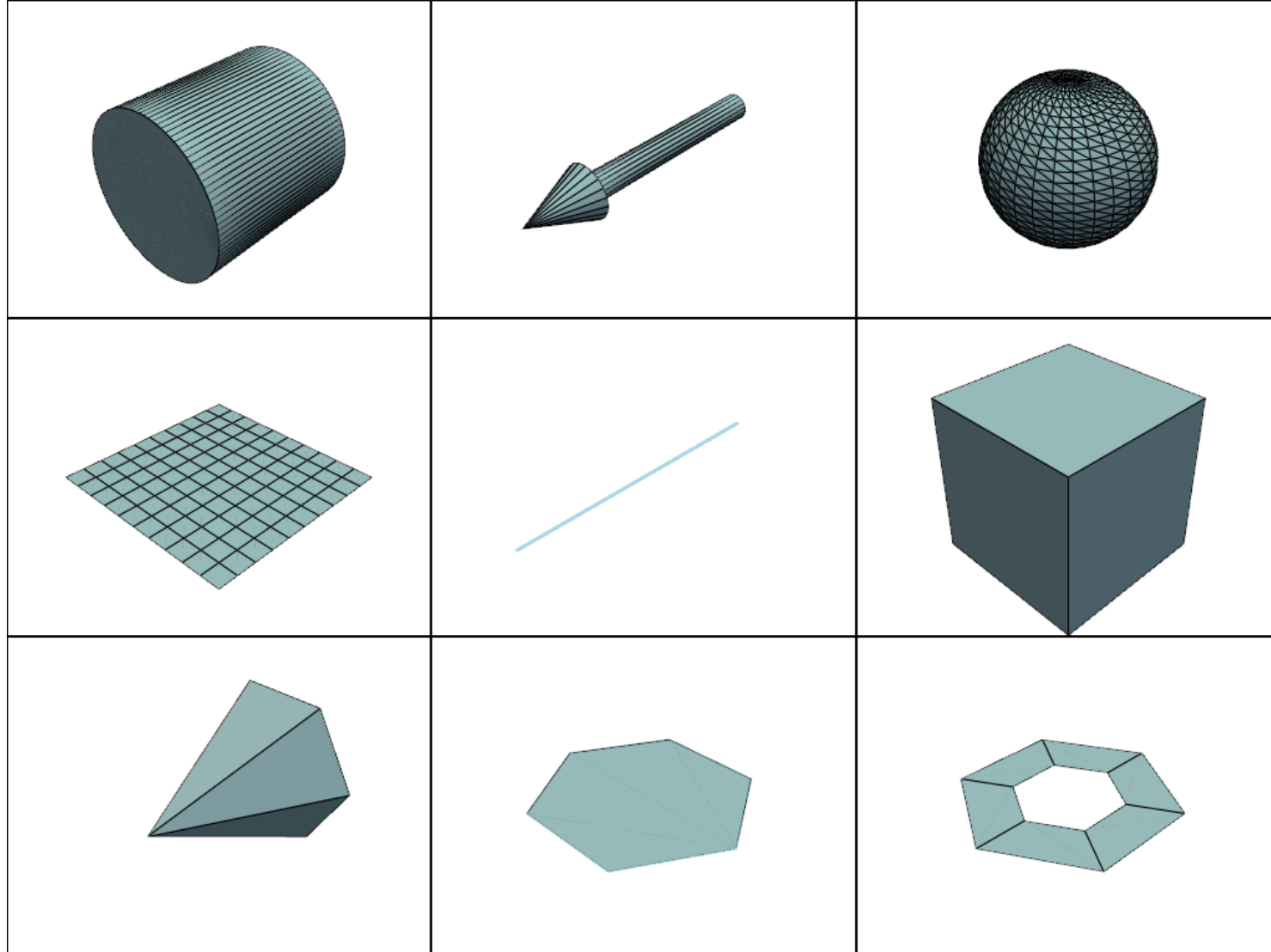
```
mesh.plot()
```

Static Scene

Interactive Scene



モデリング



モデリング

```
# Plotterオブジェクト
```

```
p1 = pv.Plotter()
```

```
# 円柱を追加します
```

```
mesh = pv.Cylinder()  
p1.add_mesh(mesh)
```

```
# 矢印を追加します
```

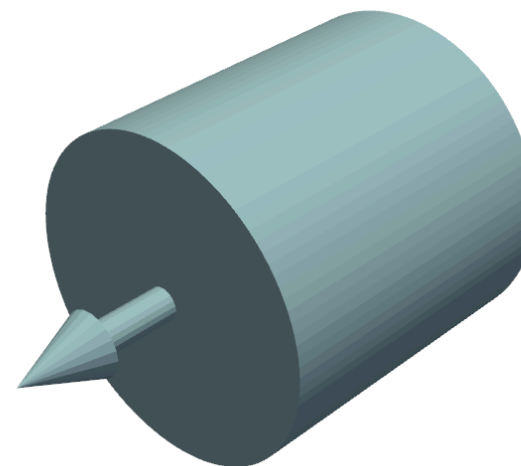
```
mesh = pv.Arrow()  
p1.add_mesh(mesh)
```

```
# 追加されたモデルを描画します
```

```
p1.show()
```

Static Scene

Interactive Scene




テクスチャ

- 概要

- PythonでCG作成

- 基礎編

- モデリング
-  テクスチャ
- ライティング
- マテリアル

- 応用編

- 空間のデータ分析
- 空間のデータ可視化

テクスチャ

テクスチャに使用する画像を読み込み

```
from pyvista import examples as ex
```

```
filename = ex.mapfile
```

画像をテクスチャとして読み込み

```
texture = pv.read_texture(filename=file
```

テクスチャをオブジェクトに貼り付け

```
mesh.plot(texture=texture)
```

Static Scene

Interactive Scene



ライティング

- 概要

- PythonでCG作成

- 基礎編

- モデリング
- テクスチャ
- 👉 ライティング
- マテリアル

- 応用編

- 空間のデータ分析
- 空間のデータ可視化

ライティング

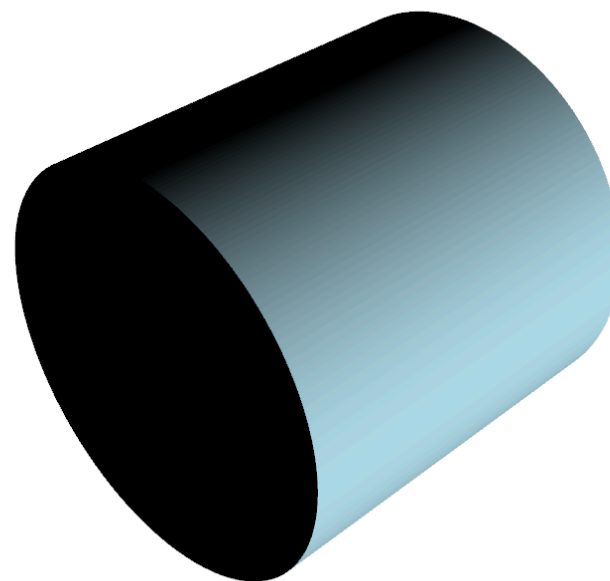
```
# Plotterクラスでlightingを無効にします。  
plotter = pv.Plotter(lighting='none')
```

```
# 仮想3D空間に光を配置します。  
light = pv.Light(  
    position=(0, 0, 1),  
    light_type='scene light'  
)
```

```
# Plotterクラスに光を追加します。  
pl.add_light(light)  
plotter.show()
```

Static Scene

Interactive Scene

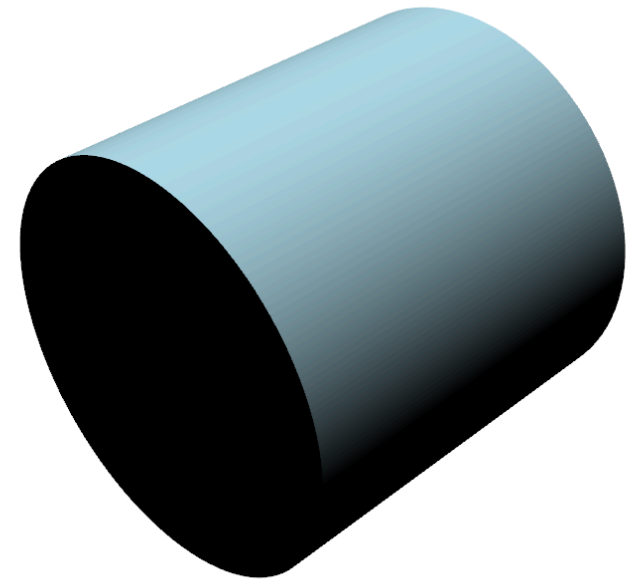


ライティング

```
# 光源の位置を(0, 0, 1)に変更します  
light = pv.Light(  
    position=(0, 0, 1),  
)
```

Static Scene

Interactive Scene



マテリアル

- 概要

- PythonでCG作成

- 基礎編

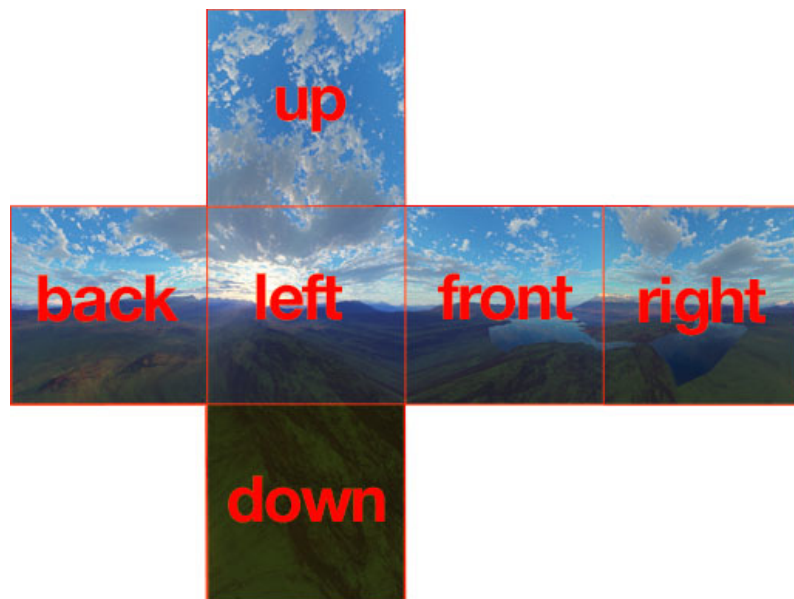
- モデリング
- テクスチャ
- ライティング
- 👉 マテリアル

- 応用編

- 空間のデータ分析
- 空間のデータ可視化

スカイボックス

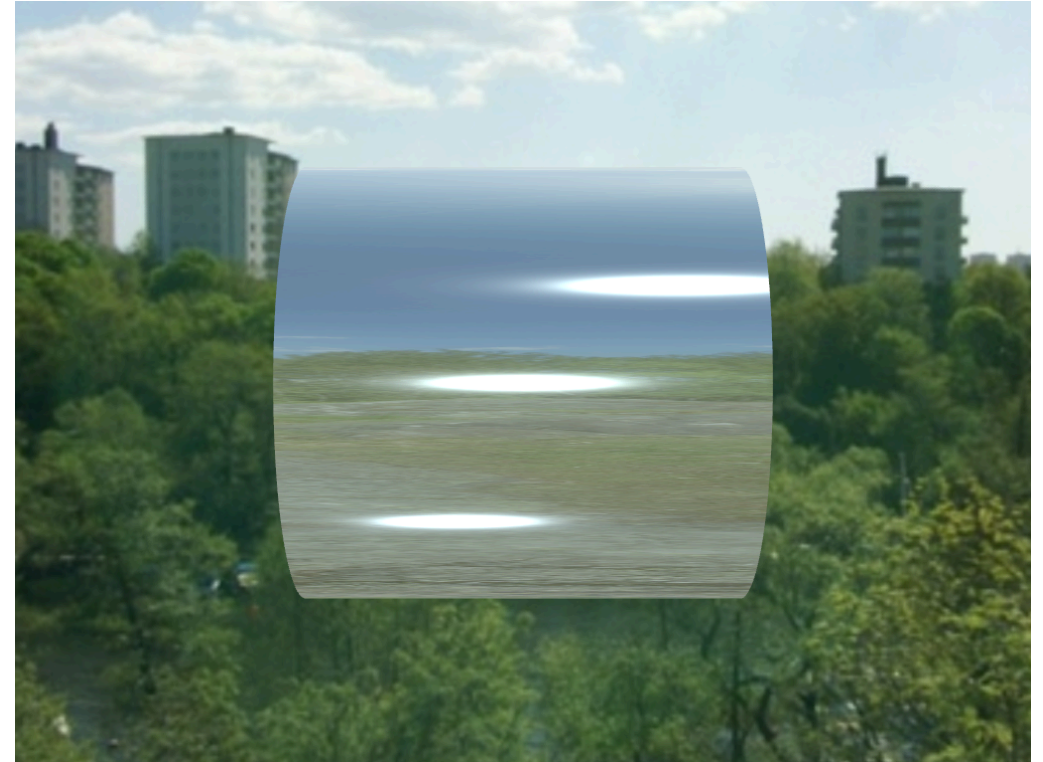
```
from pyvista import examples as ex  
  
# スカイボックスをダウンロードする  
  
cube_map = ex.download_sky_box_cube_map  
cube_map.plot()
```



質感と背景の映り込み

```
# スカイボックスを背景に設定する
pl.add_actor(cube_map.to_skybox())
# 背景の映込をテクスチャとして設定する
pl.set_environment_texture(cube_map)

# 物理ベースレンダリングを使用して
# 表面に反射する光の強さを設定する
pl.add_mesh(
    mesh,
    pbr=True,
    metallic=0.8,
    roughness=0.1,
    diffuse=1
)
```



空間のデータ分析


- 概要

- PythonでCG作成

- 基礎編

- モデリング
- テクスチャ
- ライティング
- マテリアル

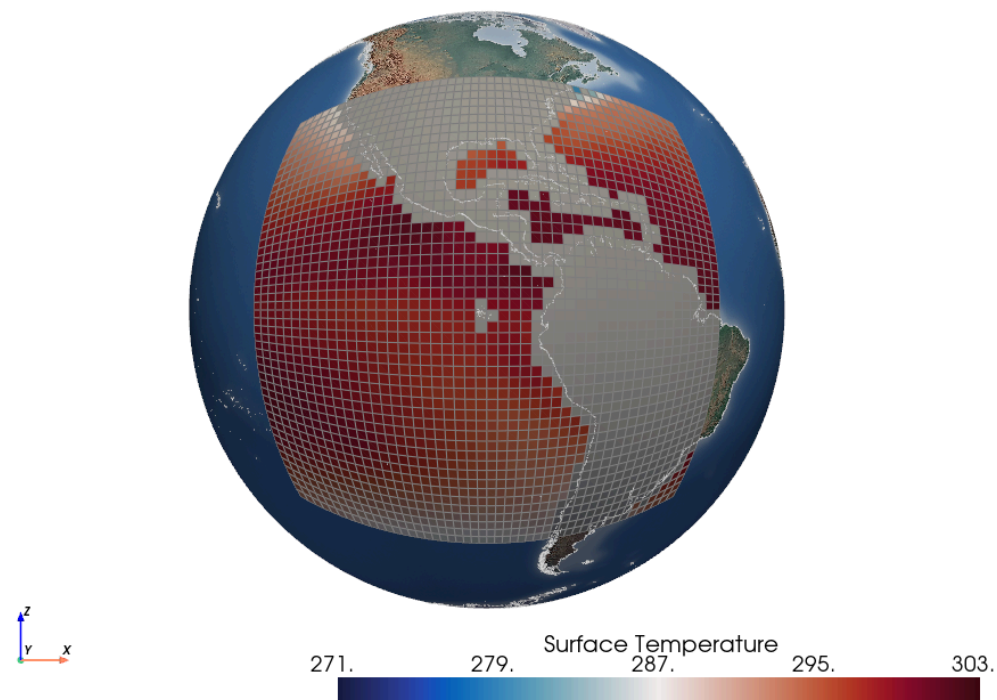
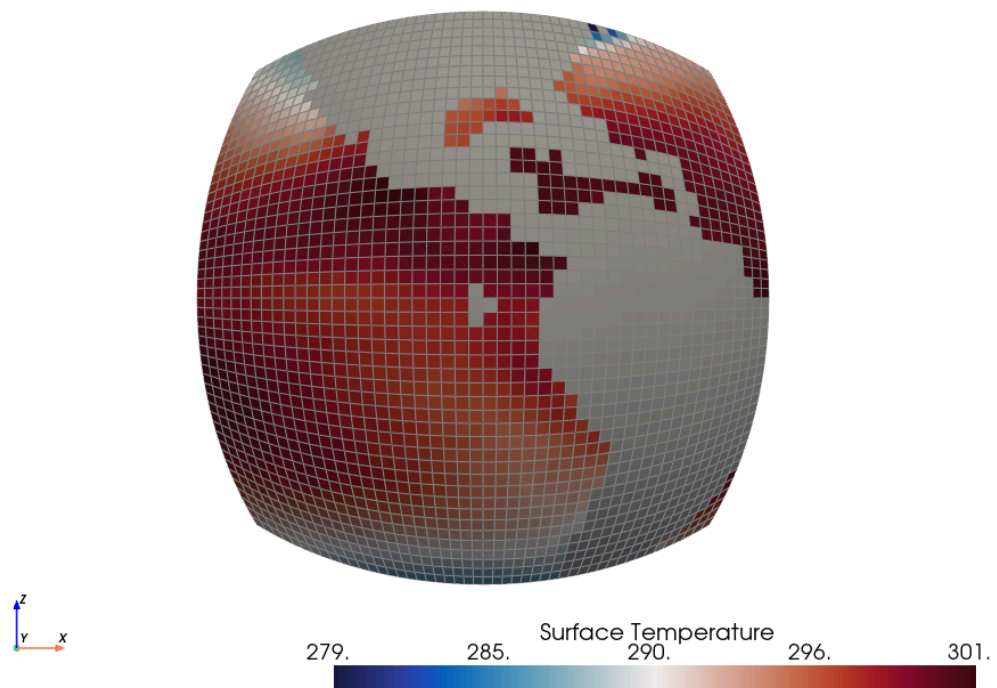
- 応用編

-  空間のデータ分析
- 空間のデータ可視化

空間のデータ分析

- Pythonを使用しているので3D空間のデータ分析も行いたい。
- PyVistaではポリゴンにデータを持たせてPandasのように処理をするメソッドが整備されている。

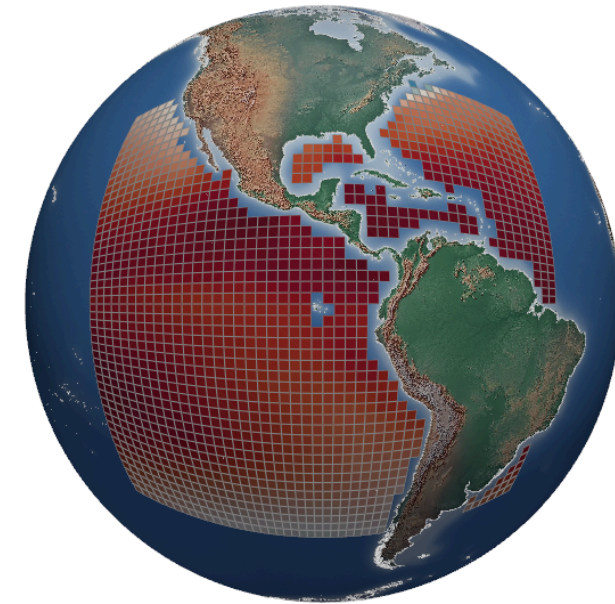
地理データの領域抽出



地理データの領域抽出

- `threshold()` メソッドは、メッシュ上のスカラー配列からnan値を持つセルを削除します。(Region Manifold Extraction)

```
# 海域の領域を抽出する  
sea_region = region.threshold()
```



空間のデータ可視化

- 概要
 - PythonでCG作成
- 基礎編
 - モデリング
 - テクスチャ
 - ライティング
 - マテリアル
- 応用編
 - 空間のデータ分析
 - 👉 空間のデータ可視化

Sphinxによる可視化

```
.. pyvista-plot::  
   :include-source: False
```

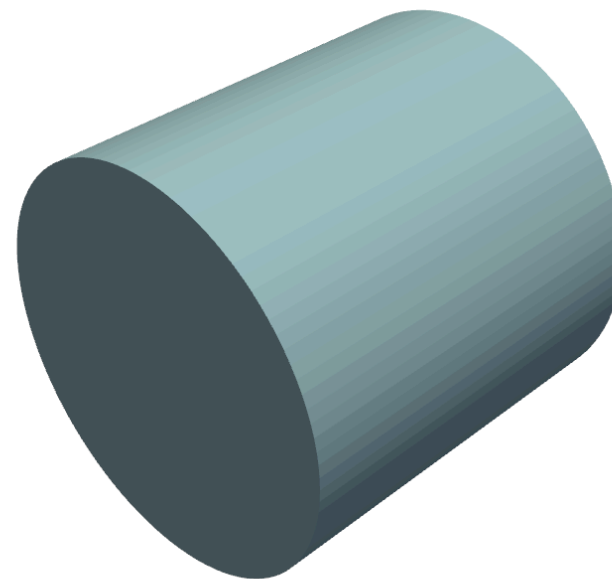
```
# pyvista-plotディレクティブを使って、  
# Sphinxドキュメントに右のような  
# 3D可視化を追加することができます。
```

```
import pyvista as pv  
mesh = pv.Cylinder()  
mesh.plot()
```

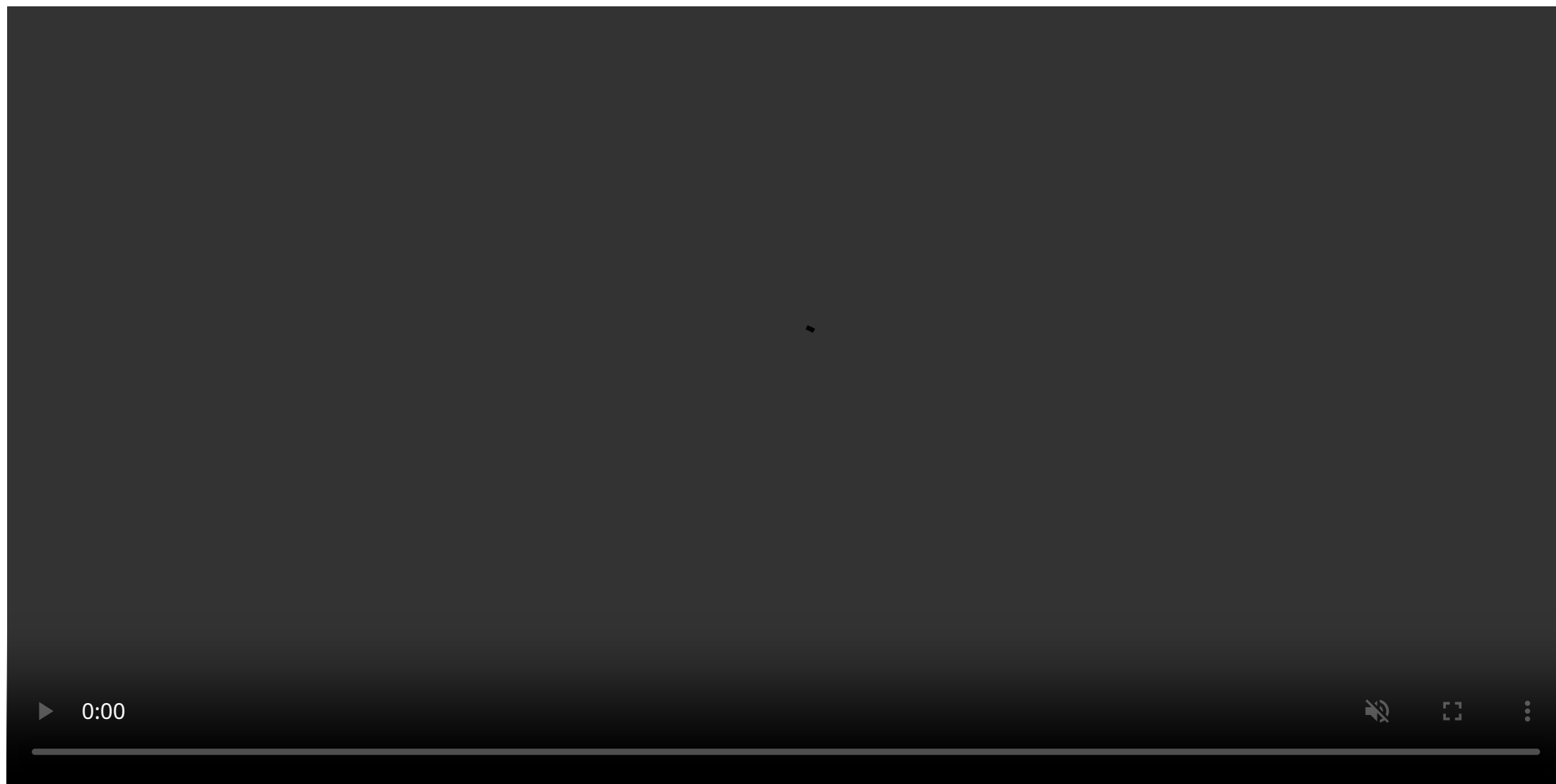
```
# このスライドもSphinxで作成しています。  
# 詳しくはsphinx-revealjsで検索！
```

Static Scene

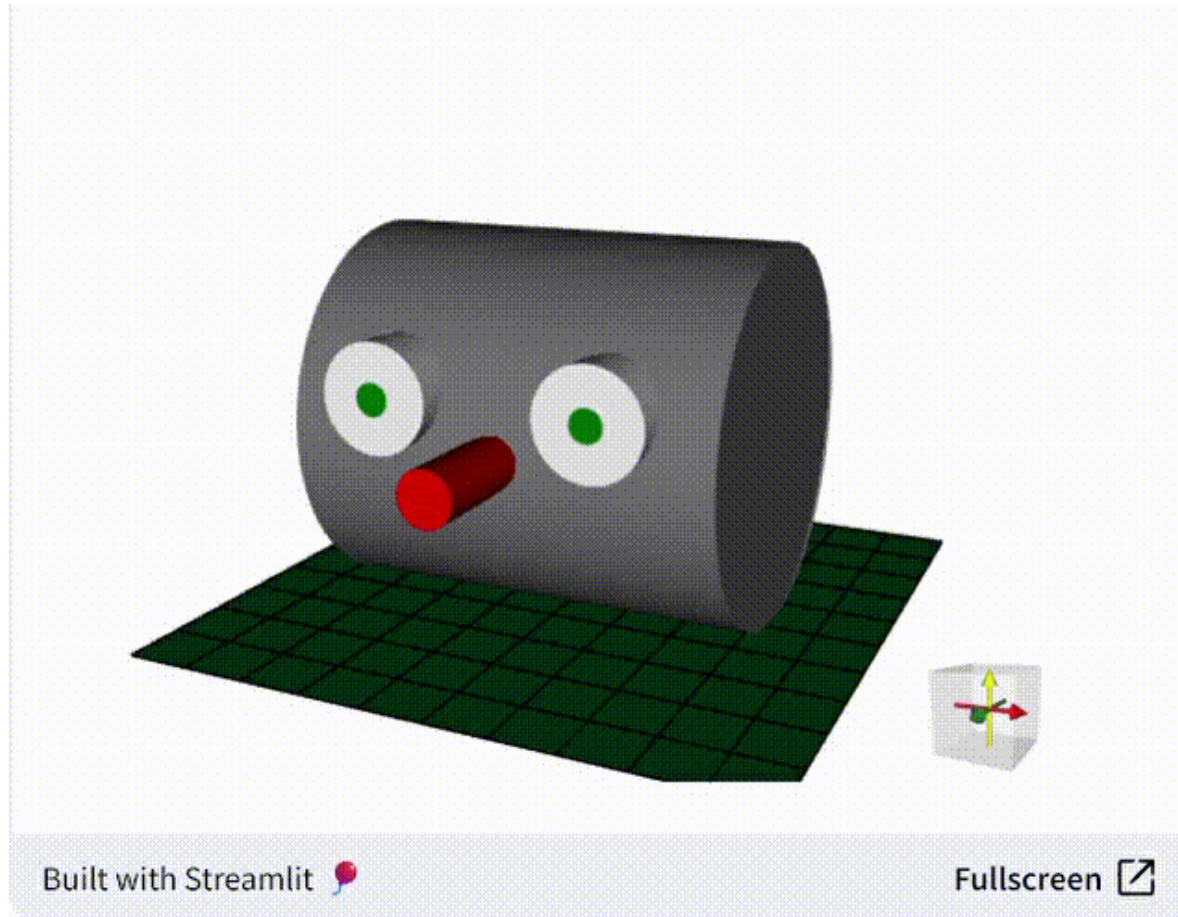
Interactive Scene



Jupyterによる可視化



Streamlitによる可視化



ご清聴ありがとうございました

☆ Star 2,648

