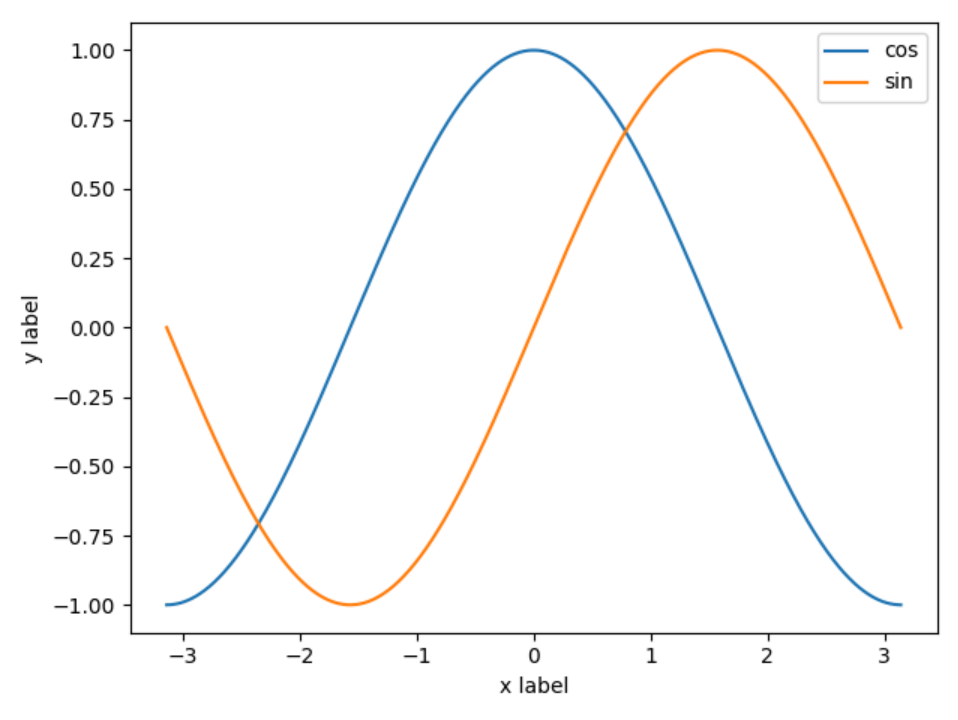
お絵描き matplotlib

概要

- matplotlib を使ってパソコンのデスクトップ画像を
  素敵なものにする
- コードをアートにも使ってみる

matplotlib

- python というプログラミング言語のパッケージの一種
- データの可視化に使用される



matplotlib の誤解

- 使いにくい
- 見た目の調整が難しい
- 見た目がダサい
- データとグラフが一緒じゃないので取り回しが悪い

誤解への回答

- データ可視化の一連の流れを覚えれば使うのはさほど難しくない
  - numpy, pandas 等と併用するのも良い
- 見た目の調整は何を弄りたいかを把握すれば簡単
  - online manual が充実している
- デフォルトの見た目は確かにいまいちなので調整しよう
- データとグラフについては運用の問題

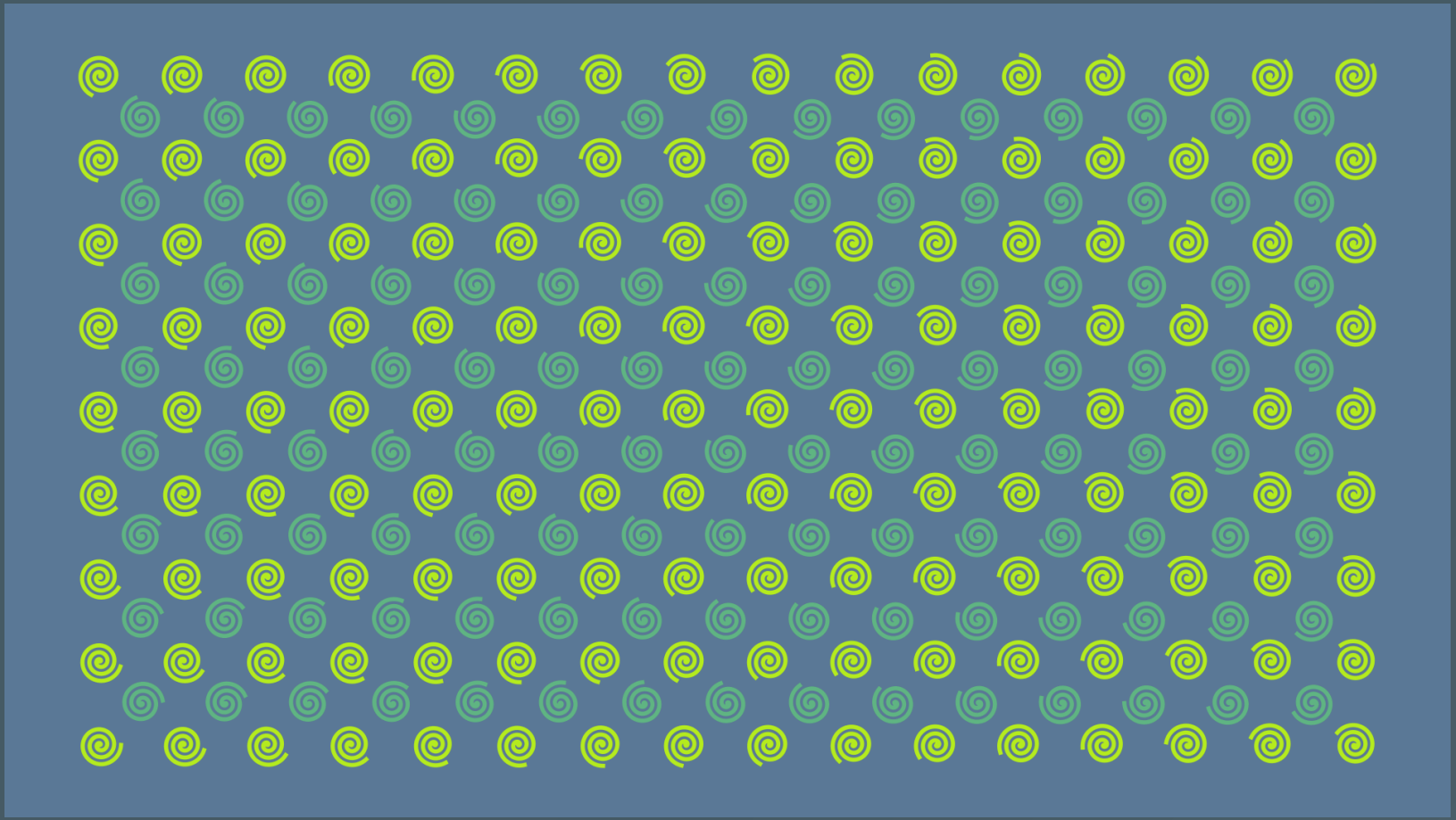
デスクトップ画像を作る

最低限の設定だけして素敵な画像を作ろう。

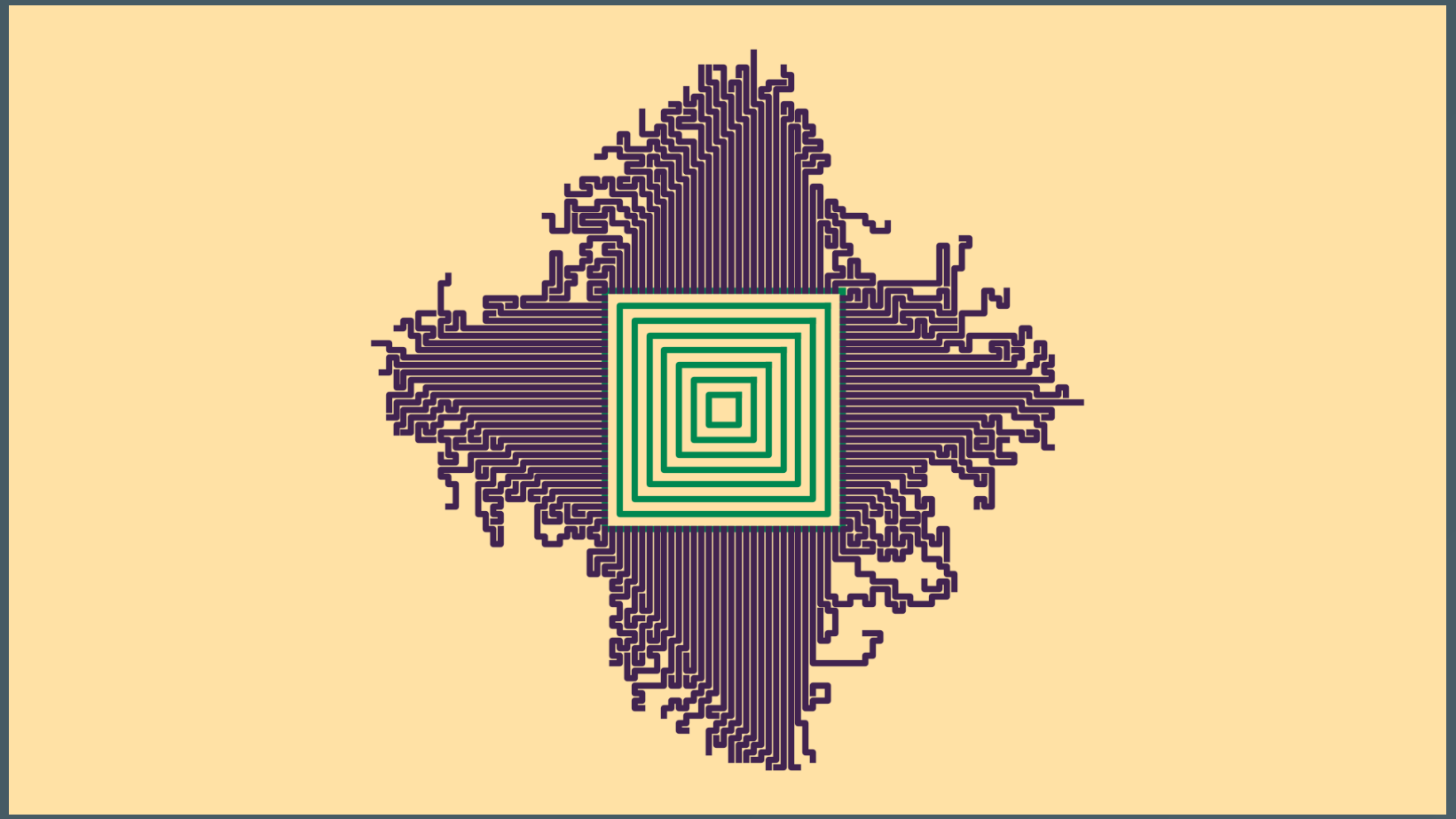
- 全体の色合いを決める
  - カラーパレットなどを参考にするとよい
- full HD などモニタの解像度に出力画像の大きさを合わせる
- どんな図形をプロットするか決める

具体例：正直角五角形

具体例：渦巻き



具体例：集積回路？



素敵な画像を作るために：１

- 解像度を変える
  - plt.rcParams["figure.dpi"] = 100
  - plt.rcParams["figure.figsize"] = [19.2, 10.8]
- 軸を消す
  - ax.axis("off")
- 背景の色を変える
  - ax.set_facecolor("#123456")
  - fig.set_facecolor("#123456")

素敵な画像を作るために：2

- プロットの色・太さを変える
  - ax.plot(x, y, color="#123456", linewidth=6)
- アスペクトを均等にする
  - ax.set_aspect("equal")
- 数学や物理などを勉強する
- etc...

コード

```python
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np


def main() -> None:
    # config
    fp: Path = Path(__file__).resolve().with_suffix(".png")
    plt.rcParams["figure.dpi"] = 100
    plt.rcParams["figure.figsize"] = [19.2, 10.8]
    c0: str = "#ff6347"
    c1: str = "#fffacd"
    c2: str = "#a52a2a"
    # generate figure
    fig: plt.Figure
    ax: plt.Axes
    fig, ax = plt.subplots()
    # plot
    n: int = 5
    t: np.ndarray = np.linspace(0, 2*np.pi, 10001, endpoint=True)
    c: np.ndarray = np.cos(t)
    s: np.ndarray = np.sin(t)
    l: float = 1/np.sqrt(1-2*np.sin(np.pi/n)*np.sin(np.pi/n))
    r: float = np.sqrt(l*l-1)
    for i in range(n):
        y: float = l*np.cos(2*np.pi/n*i)+r*c
        x: float = l*np.sin(2*np.pi/n*i)+r*s
        p: np.ndarray = np.array([[k, l] for k, l in zip(x, y) if k*k+l*l < 1])
        p = p.T
        ax.plot(p[0], p[1], color=c1, marker="o", linestyle="", markersize=54)
    ax.plot(c, s, color=c2, linewidth=60)
    # arrange
    ax.set_xlim(-1.1, 1.1)
    ax.set_ylim(-1.1, 1.1)
    ax.axis("off")
    ax.set_aspect("equal")
    ax.set_facecolor(c0)
    fig.set_facecolor(c0)
    fig.tight_layout()
    # output
    fig.savefig(fp)
    plt.cla()
    plt.clf()
    plt.close(fig)
    return None


if __name__ == "__main__":
    main()
```

コード詳細：package

```
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
```

コード詳細：config

```
fp: Path = Path(__file__).resolve().with_suffix(".png")
plt.rcParams["figure.dpi"] = 100
plt.rcParams["figure.figsize"] = [19.2, 10.8]
c0: str = "#ff6347"
c1: str = "#fffacd"
c2: str = "#a52a2a"
```

コード詳細：generate figure and axes

```
fig: plt.Figure
ax: plt.Axes
fig, ax = plt.subplots()
```

コード詳細：plot

```
    n: int = 5
    t: np.ndarray = np.linspace(0, 2*np.pi, 10001, endpoint=True)
    c: np.ndarray = np.cos(t)
    s: np.ndarray = np.sin(t)
    l: float = 1/np.sqrt(1-2*np.sin(np.pi/n)*np.sin(np.pi/n))
    r: float = np.sqrt(l*l-1)
    for i in range(n):
        y: float = l*np.cos(2*np.pi/n*i)+r*c
        x: float = l*np.sin(2*np.pi/n*i)+r*s
        p: np.ndarray = np.array([[k, l]  for k, l in zip(x, y)  if k*k+l*l < 1])
        p = p.T
        ax.plot(p[0], p[1], color=c1, marker="o", linestyle="", markersize=54)
    ax.plot(c, s, color=c2, linewidth=60)
```

コード詳細：arrange

```
    ax.set_xlim(-1.1, 1.1)
    ax.set_ylim(-1.1, 1.1)
    ax.axis("off")
    ax.set_aspect("equal")
    ax.set_facecolor(c0)
    fig.set_facecolor(c0)
    fig.tight_layout()
```

コード詳細：output

```
    fig.savefig(fp)
    plt.cla()
    plt.clf()
    plt.close(fig)
```

まとめ

- matplotlib で楽しくお絵描きができる
- 実務だけでなくアートにもコードを使ってみよう

ご清聴ありがとうございました

余談 1
本スライドは Marp を使用して作成しました。
Markdown に少し手を加えるだけで pdf にできるので非常に楽です。

余談 2
「matplotlib 展示場 1」というワールドで
今までに作成した画像の一部を展示しています。