

# 数独パズルを解いてみよう！

# 数独パズルとは？

9×9のマス目に  
1から9までの数字を、  
縦、横、3×3のブロック内  
で重複しないように配置する  
数字パズル

## 遊び方

- A タテ列に1から9の数字が入ります。
- B ヨコ列に1から9の数字が入ります。
- C 3×3のブロックに1から9の数字が入ります。

## ★ポイント

タテ列、ヨコ列、太線で囲まれた3×3のブロック内で数字が重複することはありません。

<https://www.yomiuri.co.jp/sudoku/play/>

自分でやると時間がかかる…

	9		2			6		
					9			7
1	7			5		8		
					8		7	
3		5				1		9
	2		5					
		7		1			8	6
4			9					
		3			2		1	



8	9	4	2	3	7	6	5	1
5	3	6	1	8	9	4	2	7
1	7	2	6	5	4	8	9	3
6	4	1	3	9	8	2	7	5
3	8	5	7	2	6	1	4	9
7	2	9	5	4	1	3	6	8
2	5	7	4	1	3	9	8	6
4	1	8	9	6	5	7	3	2
9	6	3	8	7	2	5	1	4

解答例

# AIに解かせてみよう！

		1		7				
2	4		1					
		3	8		9			6
6			5		4	9	1	
				1				
	1	2	6		7			8
4			7		5	2		
					1		8	9
				3		6		

AI



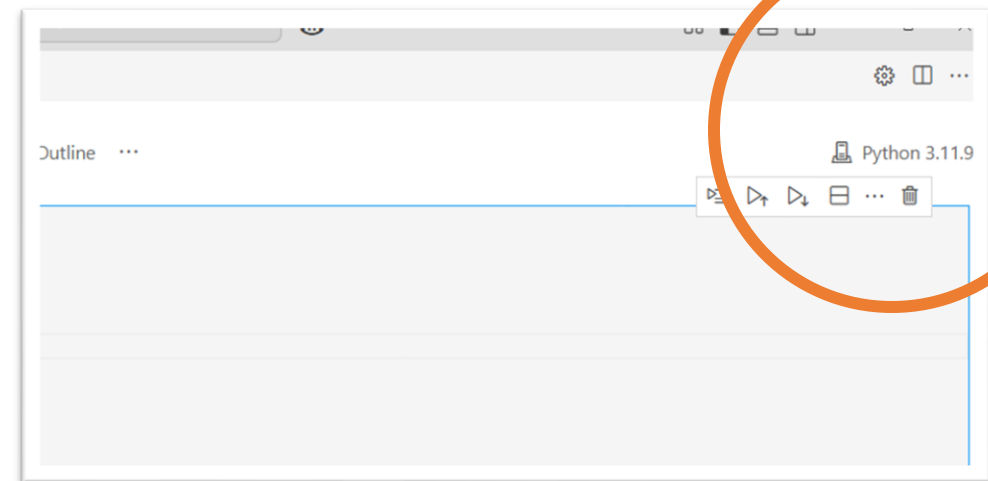
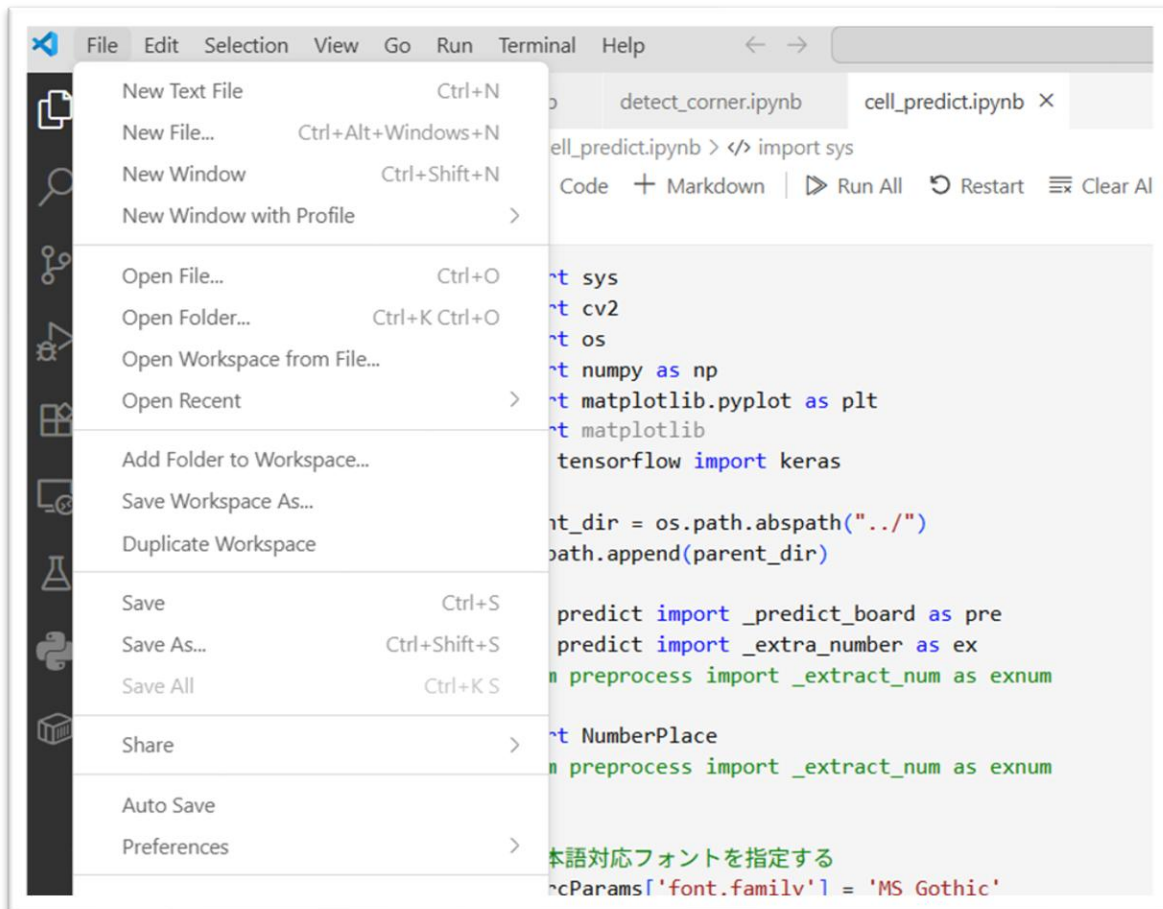
Solved

8	6	1	2	7	3	4	9	5
2	4	9	1	5	6	8	7	3
7	5	3	8	4	9	1	2	6
6	3	8	5	2	4	9	1	7
9	7	4	3	?	8	5	6	2
5	1	2	6	■	7	3	4	8
4	9	6	7	8	5	2	3	1
3	2	5	4	6	1	7	8	9
1	8	7	9	3	2	6	5	4

# プログラムの実行

1. Vscode起動
2. File → Open Folder
3. Windows → ユーザー → user → numberplace\_demo\_completed\_main

Python 3.11.9  
になっていることを確認



# プログラムの実行2

detect\_corner.ipynb

- picturesファイルの写真確認  
5枚サンプルが入っている
- detect\_corner.ipynb

画像を読み込んで、数独の盤面を切り出した新たな画像を保存する

- image○を変更すると取り出す画像を変更できる
- output○を変更すると保存する画像名を変更できる

```
import cv2
import sys
import os
import numpy as np
parent_dir = os.path.abspath("..")
sys.path.append(parent_dir)

from preprocess import _detect_corners as de
from preprocess import _trim_board as tri

# 画像を読み込む
raw_img = cv2.imread("../pictures/image1.png") #picturesフォルダにある画像を取り出す

# detect_corners に渡してコーナーを検出
corners, score = de.detect_corners(raw_img)

if corners is None:
    print("コーナー検出に失敗しました。")
    print("キャプチャからやり直してください")
else:
    print("検出された四隅座標:", corners)
    print("スコア:", score)

    corners = tri.expand_corners(corners, expand_ratio=0.02)

    trimmed = tri.trim_board(raw_img, corners) #検出した輪郭から正方形に切り出す
    cv2.imshow("trimmed", trimmed)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
    cv2.imwrite("../pictures/output1.png", trimmed) #切り出した画像をpicturesフォルダに保存する
```



# プログラムの実行3

cell\_predict.ipynb

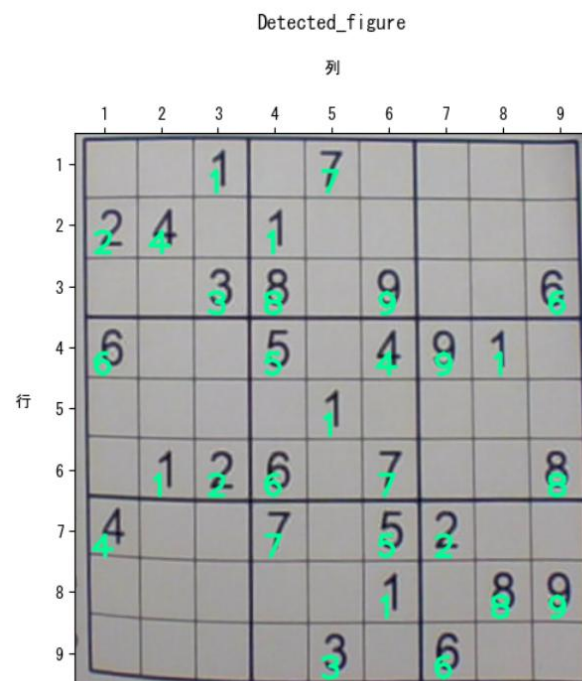
- cell\_predict.ipynb

Output0.pngを読み込んで盤面の数字を認識、数独を解く

- 認識した数字に誤りがある場合は

行、列、正しい数字の順に入力する

- 訂正箇所がない、または訂正し終わったら行選択の場面で0を入力する



```
import sys
import cv2
import os
import numpy as np
import matplotlib.pyplot as plt
import matplotlib
from tensorflow import keras

parent_dir = os.path.abspath("../")
sys.path.append(parent_dir)

from predict import _predict_board as pre
from predict import _extra_number as ex
#from preprocess import _extract_num as exnum

import NumberPlace
#from preprocess import _extract_num as exnum

#日本語対応フォントを指定する
plt.rcParams['font.family'] = 'MS Gothic'

img_path = "../pictures/output1.png" #コーナー検出済みの画像をpicturesフォルダから読み込む
```

# やってみよう

camera\_start.ipynbに、PCのカメラを起動し写真を撮るプログラムを作成して、自分の数独を解いてみよう

```
import cv2

cv2.namedWindow("camera", cv2.WINDOW_NORMAL)

#カメラの設定 デバイスIDは0
cap = cv2.VideoCapture(0)

#繰り返しのためのwhile文
while True:
    #カメラからの画像取得
    ret, frame = cap.read()

    #カメラの画像の出力
    cv2.imshow('camera' , frame)

    #繰り返し分から抜けるためのif文
    key =cv2.waitKey(10)
    if key == 27: #escapeで終了
        break

    elif key == 32: # スペースキーで画像保存
        filename = "../pictures/image9.png"
        cv2.imwrite(filename, frame)
        print(f"{filename} を保存しました。")

#メモリを解放して終了するためのコマンド
cap.release()
cv2.destroyAllWindows()
```