

# GISプログラミング入門

オープンソース地理空間ツールのための実践的Pythonガイド

Qiusheng Wu



# GIS プログラミング 入門

オープンソース地理空間ツールのための実践  
的 Python ガイド

Qiusheng Wu  
2025

# Contents

<b>序文</b>	<b>1</b>
はじめに	3
この本の対象読者	3
この本の内容	4
この本を最大限活用するために	5
この本で使用する慣例	5
コード例のダウンロード	6
ビデオチュートリアル	6
お問い合わせ	7
謝辞	7
著者について	8
ライセンスと著作権	8
<b>I: ソフトウェアの設定</b>	<b>11</b>
<b>1. ソフトウェアツールの概要</b>	<b>13</b>
1.1. はじめに	13
1.2. 学習目標	13
1.3. 必須ソフトウェアツール	13
1.4. ツール統合とワークフロー	15
1.5. コード例の実行	16
1.6. 重要なポイント	16
<b>2. Python パッケージ管理入門</b>	<b>17</b>
2.1. はじめに	17
2.2. 学習目標	17
2.3. Conda (Miniconda) のインストール	18
2.4. Conda の概念を理解する	20
2.5. 最初の地理空間環境の作成	20
2.6. Conda のトラブルシューティング	21
2.7. 必須 Conda コマンド	22
2.8. uv: 高速な代替品の紹介	27
2.9. パッケージ管理のベストプラクティス	28
2.10. 重要なポイント	29
2.11. 演習	29
<b>3. Visual Studio Code のセットアップ</b>	<b>31</b>
3.1. はじめに	31
3.2. 学習目標	31
3.3. Visual Studio Code のインストール	32
3.4. Python プログラミングのための必須拡張機能	32
3.5. Python 開発用の VS Code 設定	34
3.6. 必須キーボードショートカット	36
3.7. 参考文献と追加学習	37
3.8. 重要なポイント	38
3.9. 演習	38
<b>4. Git によるバージョン管理</b>	<b>40</b>

4.1.	はじめに	40
4.2.	学習目標	41
4.3.	GitHub アカountのセットアップ	41
4.4.	Git のインストール	41
4.5.	Git の設定	42
4.6.	Git 概念の理解	43
4.7.	必須 Git コマンド	43
4.8.	GitHub の使用	47
4.9.	VS Code との統合	47
4.10.	地理空間プロジェクトのベストプラクティス	48
4.11.	重要なポイント	49
4.12.	演習	49
5.	<b>Google Colab の使用</b>	<b>51</b>
5.1.	はじめに	51
5.2.	学習目標	51
5.3.	Google Colab の開始	51
5.4.	地理空間環境のセットアップ	53
5.5.	必須 Colab 機能	54
5.6.	Colab でコード例を実行	56
5.7.	重要なポイント	56
5.8.	演習	56
6.	<b>JupyterLab での作業</b>	<b>58</b>
6.1.	はじめに	58
6.2.	学習目標	58
6.3.	JupyterLab のインストールとセットアップ	58
6.4.	JupyterLab の開始	59
6.5.	必須キーボードショートカット	62
6.6.	MyBinder でのコード例実行	64
6.7.	重要なポイント	64
6.8.	演習	65
7.	<b>Docker の使用</b>	<b>67</b>
7.1.	はじめに	67
7.2.	学習目標	67
7.3.	Docker Desktop のインストール	67
7.4.	基本概念	69
7.5.	Docker でコード例を実行する	70
7.6.	一般的な Docker コマンド	71
7.7.	重要なポイント	72
7.8.	演習	73
II:	<b>Python プログラミングの基礎</b>	<b>75</b>
8.	<b>変数とデータ型</b>	<b>77</b>
8.1.	はじめに	77
8.2.	学習目標	77
8.3.	Python の変数	77
8.4.	変数の命名	78

8.5. データ型 .....	79
8.6. エスケープ文字 .....	80
8.7. Python のコメント .....	80
8.8. 変数とデータ型の操作 .....	80
8.9. 基本的な文字列操作 .....	81
8.10. 重要なポイント .....	82
8.11. 演習 .....	82
<b>9. Python データ構造 .....</b>	<b>84</b>
9.1. はじめに .....	84
9.2. 学習目標 .....	84
9.3. タプル .....	84
9.4. リスト .....	85
9.5. セット .....	88
9.6. 辞書 .....	90
9.7. データ構造選択ガイド .....	93
9.8. 重要なポイント .....	94
9.9. 演習 .....	94
<b>10. 文字列操作 .....</b>	<b>96</b>
10.1. はじめに .....	96
10.2. 学習目標 .....	96
10.3. 文字列の作成と操作 .....	96
10.4. 地理空間データのための文字列メソッド .....	98
10.5. 文字列フォーマット .....	101
10.6. 文字列操作判断ガイド .....	104
10.7. 重要なポイント .....	105
10.8. 演習 .....	105
<b>11. ループと条件文 .....</b>	<b>107</b>
11.1. はじめに .....	107
11.2. 学習目標 .....	107
11.3. For ループ .....	107
11.4. While ループ .....	109
11.5. 制御文：コード内での決定 .....	110
11.6. ループと制御文の組み合わせ .....	112
11.7. ループと制御文の決定ガイド .....	113
11.8. 重要なポイント .....	114
11.9. 演習 .....	114
<b>12. 関数とクラス .....</b>	<b>116</b>
12.1. はじめに .....	116
12.2. 学習目標 .....	116
12.3. 関数：再利用可能なコードブロックの構築 .....	116
12.4. クラス：データと動作の統合 .....	121
12.5. 関数とクラスの組み合わせ .....	123
12.6. 関数とクラスの設計ガイドライン .....	124
12.7. 重要なポイント .....	124
12.8. 演習 .....	125

<b>13. ファイルの操作</b> .....	<b>126</b>
13.1. はじめに .....	126
13.2. 学習目標 .....	126
13.3. サンプルファイルの作成 .....	126
13.4. ファイルの読み取りと書き込み .....	127
13.5. 例外処理 .....	129
13.6. ファイル処理と例外処理の組み合わせ .....	130
13.7. 異なるファイル形式の操作 .....	132
13.8. 重要なポイント .....	133
13.9. 演習 .....	134
<b>14. NumPy と Pandas によるデータ分析</b> .....	<b>137</b>
14.1. はじめに .....	137
14.2. 学習目標 .....	137
14.3. NumPy の紹介 .....	138
14.4. Pandas の紹介 .....	148
14.5. NumPy と Pandas の組み合わせ .....	155
14.6. 重要なポイント .....	156
14.7. 参考文献 .....	157
14.8. 演習 .....	157
<b>III: Python による地理空間プログラミング</b> .....	<b>159</b>
<b>15. 地理空間 Python の紹介</b> .....	<b>161</b>
15.1. はじめに .....	161
15.2. 地理空間 Python エコシステム .....	161
15.3. ライブラリの関係の理解 .....	162
15.4. 環境の設定 .....	162
15.5. 検証と最初のステップ .....	164
15.6. 学習パスと章の概要 .....	164
15.7. 覚えておくべき主要概念 .....	165
15.8. ヘルプとリソースの入手 .....	166
15.9. 次のステップ .....	166
15.10. 演習 .....	166
<b>16. GeoPandas によるベクトルデータ分析</b> .....	<b>168</b>
16.1. はじめに .....	168
16.2. 学習目標 .....	168
16.3. 中核概念 .....	169
16.4. GeoPandas のインストール .....	169
16.5. GeoDataFrame の作成 .....	169
16.6. 地理空間データの読み書き .....	170
16.7. 投影と座標参照系 (CRS) .....	172
16.8. 空間測定と解析 .....	173
16.9. 地理空間データの可視化 .....	176
16.10. 高度な幾何学的操作 .....	180
16.11. 空間関係とクエリ .....	184
16.12. ベストプラクティスとパフォーマンスの考慮事項 .....	185
16.13. 主要なポイント .....	185

16.14. 演習 .....	186
<b>17. Rasterio を使ったラスターデータの操作 .....</b>	<b>187</b>
17.1. はじめに .....	187
17.2. 学習目標 .....	187
17.3. Rasterio のインストール .....	188
17.4. ラスタデータの読み取り .....	188
17.5. ラスタデータの可視化 .....	191
17.6. ラスタバンドのアクセスと操作 .....	199
17.7. ラスタデータの書き込み .....	201
17.8. ラスタデータのクリッピング .....	202
17.9. 主要なポイント .....	205
17.10. 演習 .....	205
<b>18. Xarray による多次元データ分析 .....</b>	<b>208</b>
18.1. はじめに .....	208
18.2. 学習目標 .....	208
18.3. Xarray のデータモデルの理解 .....	209
18.4. 環境の設定 .....	210
18.5. 実際の気候データの読み込みと探索 .....	210
18.6. DataArray の操作 .....	212
18.7. 直感的なデータ選択とインデクシング .....	214
18.8. 多次元データでの操作実行 .....	215
18.9. Xarray によるデータ可視化 .....	216
18.10. データセットでの作業：複数の変数 .....	219
18.11. ラベルベース操作の力 .....	220
18.12. 高度なインデクシング技術 .....	221
18.13. 高レベル計算操作 .....	222
18.14. データの入出力 .....	225
18.15. 主要なポイント .....	226
18.16. さらなる読み物 .....	227
18.17. 演習 .....	227
<b>19. Rioxarray によるラスター分析 .....</b>	<b>229</b>
19.1. はじめに .....	229
19.2. 学習目標 .....	229
19.3. Rioxarray 環境の設定 .....	229
19.4. 地理参照ラスターデータの読み込みと探索 .....	230
19.5. 基本的な地理空間操作 .....	233
19.6. 空間次元と解像度の操作 .....	235
19.7. 地理空間ラスターデータの可視化 .....	237
19.8. データストレージとファイル管理 .....	240
19.9. 座標系の比較 .....	242
19.10. バンド演算入門 .....	244
19.11. 重要なポイント .....	248
19.12. 演習 .....	248
<b>20. Leafmap によるインタラクティブ可視化 .....</b>	<b>251</b>
20.1. はじめに .....	251

20.2.	学習目標	252
20.3.	Leafmap のインストールとセットアップ	252
20.4.	インタラクティブマップの作成	253
20.5.	ベースマップの変更	256
20.6.	ベクターデータの可視化	261
20.7.	コロプレスマップの作成	267
20.8.	GeoParquet データの可視化	268
20.9.	PMTiles の可視化	270
20.10.	ラスターデータの可視化	274
20.11.	Maxar Open Data のアクセスと可視化	281
20.12.	主要なポイント	287
20.13.	演習	288
<b>21.</b>	<b>WhiteboxTools による地理処理</b>	<b>291</b>
21.1.	はじめに	291
21.2.	学習目標	291
21.3.	なぜ Whitebox なのか?	291
21.4.	Whitebox の有用なリソース	293
21.5.	Whitebox のインストール	293
21.6.	流域解析	294
21.7.	LiDAR データ解析	307
21.8.	重要なポイント	314
21.9.	演習問題	315
<b>22.</b>	<b>MapLibre による 3D マッピング</b>	<b>318</b>
22.1.	はじめに	318
22.2.	学習目標	318
22.3.	有用なリソース	318
22.4.	インストールとセットアップ	318
22.5.	インタラクティブマップの作成	319
22.6.	マップコントロールの追加	320
22.7.	レイヤーの追加	323
22.8.	MapTiler の使用	325
22.9.	3D マッピング	326
22.10.	ベクターデータの視覚化	333
22.11.	ラスターデータの視覚化	343
22.12.	カスタムコンポーネントの追加	345
22.13.	PMTiles の視覚化	352
22.14.	DeckGL レイヤーの追加	357
22.15.	HTML へのエクスポート	360
22.16.	重要なポイント	361
22.17.	演習問題	361
<b>23.</b>	<b>Earth Engine と Geemap によるクラウドコンピューティング</b>	<b>363</b>
23.1.	はじめに	363
23.2.	学習目標	363
23.3.	Google Earth Engine の紹介	363
23.4.	インタラクティブマップとツールの紹介	366



23.5.	Earth Engine データカタログ	372
23.6.	Earth Engine データタイプ	374
23.7.	Earth Engine ラスターデータ	374
23.8.	Earth Engine ベクターデータ	377
23.9.	Earth Engine データ可視化のためのその他のツール	380
23.10.	ベクターデータ処理	389
23.11.	ラスターデータ処理	391
23.12.	Earth Engine データのエクスポート	398
23.13.	タイムラプスアニメーションの作成	401
23.14.	Earth Engine データのチャート作成	406
23.15.	主要な要点	435
23.16.	演習	436
<b>24.</b>	<b>HyperCoast によるハイパースペクトラルデータ可視化</b>	<b>438</b>
24.1.	はじめに	438
24.2.	学習目標	438
24.3.	環境セットアップ	439
24.4.	ハイパースペクトルデータの検索	439
24.5.	ハイパースペクトルデータのダウンロード	442
24.6.	ハイパースペクトルデータの読み取り	442
24.7.	ハイパースペクトルデータの可視化	443
24.8.	画像キューブの作成	444
24.9.	インタラクティブスライシング	446
24.10.	インタラクティブ閾値処理	447
24.11.	主要な要点	448
24.12.	演習問題	449
<b>25.</b>	<b>DuckDB による高性能地理空間分析</b>	<b>451</b>
25.1.	はじめに	451
25.2.	学習目標	451
25.3.	インストールとセットアップ	452
25.4.	空間分析のための SQL 基礎	454
25.5.	Python API 統合	459
25.6.	データインポート	461
25.7.	データエクスポート	465
25.8.	ジオメトリとの作業	467
25.9.	空間関係	471
25.10.	空間結合	473
25.11.	大規模データ分析	476
25.12.	主要な要点	483
25.13.	演習問題	484
<b>26.</b>	<b>GDAL と OGR による地理空間データ処理</b>	<b>488</b>
26.1.	はじめに	488
26.2.	学習目標	489
26.3.	インストールとセットアップ	489
26.4.	サンプルデータセット	489
26.5.	データの理解	490

26.6.	座標変換 .....	490
26.7.	形式変換 .....	491
26.8.	クリッピングとマスキング .....	492
26.9.	ラスター分析と計算 .....	493
26.10.	ラスターとベクター間の変換 .....	495
26.11.	ジオメトリ処理 .....	495
26.12.	フィールドとレイヤーの管理 .....	496
26.13.	タイリングとデータ管理 .....	497
26.14.	高度なラスター処理 .....	499
26.15.	地形分析 .....	500
26.16.	主要な要点 .....	506
26.17.	参考文献と参考資料 .....	507
26.18.	演習問題 .....	507
<b>27.</b>	<b>Voilà と Solara によるインタラクティブダッシュボードの構築 .....</b>	<b>510</b>
27.1.	はじめに .....	510
27.2.	学習目標 .....	511
27.3.	Voilà と Solara のインストール .....	511
27.4.	Hugging Face Spaces の紹介 .....	512
27.5.	基本的な Voilà アプリケーションの作成 .....	512
27.6.	Solara による高度な Web アプリケーションの作成 .....	518
27.7.	主要な要点 .....	524
27.8.	演習問題 .....	525
<b>28.</b>	<b>Apache Sedona による分散コンピューティング .....</b>	<b>526</b>
28.1.	はじめに .....	526
28.2.	学習目標 .....	527
28.3.	Apache Sedona のインストールとセットアップ .....	527
28.4.	サンプルデータのダウンロード .....	529
28.5.	コア概念とデータ構造 .....	530
28.6.	空間操作と関数 .....	532
28.7.	空間結合とインデックス .....	535
28.8.	高度な空間分析 .....	538
28.9.	ベクターデータの読み取り .....	540
28.10.	ベクターデータの可視化 .....	543
28.11.	ベクターデータの書き込み .....	546
28.12.	ラスターデータの読み取り .....	546
28.13.	ラスターデータの可視化 .....	548
28.14.	ラスターマップ代数 .....	549
28.15.	ラスターゾーン統計 .....	550
28.16.	ラスターデータの書き込み .....	551
28.17.	GeoPandas との統合 .....	552
28.18.	実世界の使用例 .....	555
28.19.	主要な要点 .....	557
28.20.	参考文献と参考資料 .....	558
28.21.	演習問題 .....	558

# 序文



## はじめに

地理情報システム (GIS) と地理空間分析は、環境科学や都市計画からビジネス分析や公衆衛生まで、数多くの分野において基本的なツールとなっています。地理空間データの量と複雑さが指数関数的に増加し続ける中、このデータをプログラムで処理、分析、可視化する能力は、空間情報を扱う研究者、アナリスト、専門家にとって必須のスキルとなっています。

Python は地理空間分析における主要なプログラミング言語として台頭し、複雑な空間操作を初心者から専門家まで利用しやすくする豊富なライブラリとツールのエコシステムを提供しています。しかし、Python 初心者から自信を持った地理空間プログラマーになる道のりは、学ぶべき数多くのライブラリと習得すべき概念があり、困難に思えるかもしれません。

この本は、Python を使った地理空間プログラミングの学習に対して、体系的で実践的なアプローチを提供することで、そのギャップを埋めています。最初から高度な技術で圧倒するのではなく、地理空間プログラミングの旅路全体を通じて役立つ必須スキルの強固な基盤を築くことに焦点を当てています。各章は前の章の上に構築され、理論的理解と実践的専門知識の両方を確実に身につけられるようになっています。

この本で採用しているアプローチは実践的で例に基づいています。実際の地理空間データセットを扱い、実用的な問題を解決し、地理空間分析と可視化における Python の力を実証するプロジェクトを構築します。この本を読み終える頃には、あなた自身の地理空間プログラミングの課題に取り組む自信とスキルを身につけているでしょう。

## この本の対象読者

この本は、地理空間分析と可視化に Python の力を活用したい多様な学習者に向けて設計されています：

**学生と研究者：**地理学、環境科学、都市計画、データサイエンス、および関連分野で、研究や学習の一部として空間データを分析する必要がある方。事前のプログラミング経験は前提としていませんが、コンピュータとデータ分析の概念に基本的な知識があると役立ちます。

**GIS 専門家：**現在デスクトップ GIS ソフトウェアを使用しており、プログラミングスキルでツールキットを拡張したい方。ArcGIS、QGIS、または類似のツールで作業し、ワークフローを自動化したり、従来の GIS ソフトウェアでは困難な分析を実行したい場合、この本がその移行を支援します。

**データサイエンティストとアナリスト：**位置ベースのデータを扱い、スキルセットに空間分析能力を追加したい方。Python の基礎に慣れているが地理空間概念が初めての場合、この本が必要な空間的思考とツールを紹介します。

**ソフトウェア開発者：**地理空間データを扱うアプリケーション構築に興味がある方。Web マッピングアプリケーション、位置機能付きモバイルアプリ、またはデータ処理パイプラインを開発している場合、この本が必要な基盤を提供します。

**独学者とキャリア転換者：**成長している地理空間データサイエンス分野に興味がある方。この本は Python プログラミングと GIS 概念のどちらも事前知識を前提としないため、意欲的な初心者にとって利用しやすくなっています。

**政府と産業の専門家：**都市計画者、環境コンサルタント、市場研究者、物流コーディネーター、公衆衛生担当者など、業務に空間分析を組み込む必要がある方。

重要な要件は好奇心と学習への意欲です。プログラミング経験があると役立ちますが、必須ではありません。基礎から始めて体系的に構築していきます。

## この本の内容

この本は、ソフトウェアセットアップから Python の基礎、高度な地理空間プログラミングまで段階的に進める 3 つのセクションに構成されています：

**ソフトウェアセットアップ**では、地理空間プログラミングに必要なすべてのツールを含む開発環境を準備します。パッケージ管理のための Miniconda、開発のための VS Code、バージョン管理のための Git、および Google Colab や JupyterLab などのクラウドベースの代替手段のインストールと設定方法を学びます。このセクションは、プログラミングに取り組む前に確固たる基盤を確保します。

**Python プログラミング基礎**では、7 つの包括的な章を通じてコアプログラミングスキルを構築します。Python の基礎から始まり、変数とデータ型、データ構造（リスト、辞書、セット）、文字列操作、ループと条件文による制御フロー、関数とクラス、ファイル処理、および NumPy と Pandas を使ったデータ分析をマスターします。これらのスキルは、すべての地理空間プログラミングタスクの基盤を形成します。

**Python での GIS プログラミング**は、自信を持った地理空間プログラマーに変身させる 14 の専門章で構成されています：

- **地理空間 Python の紹介** - 核となる概念と Python 地理空間エコシステム
- **GeoPandas によるベクターデータ分析** - 点、線、ポリゴンの操作
- **Rasterio によるラスターデータ** - 衛星画像とグリッドデータセットの処理
- **Xarray による多次元データ分析** - 複雑な科学データセットの処理
- **Rioxarray によるラスター分析** - 高度なラスター処理と分析
- **Leafmap によるインタラクティブ可視化** - 動的で対話的な地図の作成
- **WhiteboxTools によるジオプロセッシング** - 高度な空間分析操作
- **MapLibre による 3D マッピング** - 三次元可視化の構築
- **Earth Engine と Geemap によるクラウド計算** - 大規模分析のための Google Earth Engine の活用
- **HyperCoast によるハイパースペクトラルデータ可視化** - ハイパースペクトラルデータの操作
- **DuckDB による高性能地理空間分析** - 高性能空間データ処理
- **GDAL と OGR による地理空間データ処理** - 様々な地理空間データ形式の操作
- **Solara によるインタラクティブダッシュボードの構築** - 地理空間アプリケーションのためのインタラクティブダッシュボード作成
- **Apache Sedona による分散計算** - 分散環境での大規模地理空間データセットの処理

各章は一貫した構造に従っています：

- 実世界の文脈での明確な概念説明
- 詳細な注釈付きの段階的コード例
- 本物の地理空間データセットを使った実践的演習
- よくある落とし穴とトラブルシューティングガイド
- 追加リソースと参考文献

進行は慎重に設計されており、各章が前の概念の上に構築されながら新しい機能を導入し、地理空間プログラミングの幅と深さの両方を確実に身につけられるようになっています。

## この本を最大限活用するために

この本での学習体験を最大化するために、以下の推奨事項を検討してください：

**適切な開発環境の設定：**本の最初のセクションで説明されているように、Python と必要なライブラリをインストールしてください。適切に設定された環境は、学習の旅路を通じて時間と苛立ちを節約します。地理空間ライブラリのインストールを簡単にするため、conda や uv を使って Python パッケージを管理することを検討してください。

**コード例に従って実行：**この本はインタラクティブに設計されています。コードを読むだけでなく、実際に入力し、実行し、修正を試してください。理解は練習を通じて得られ、各例は後で必要になるスキルを構築します。

**演習に取り組む：**各章には、学んだ概念を強化するために設計された演習が含まれています。これらはオプションの付加ではなく、学習プロセスの不可欠な部分です。ガイド付き演習から始めて、その後自分自身のプロジェクトに挑戦してください。

**実際のデータを使用：**本では例と演習のためのデータセットを提供していますが、自分の分野や興味のあるデータに技術を適用してみてください。これは概念が実世界のシナリオにどのように適用されるかを理解し、能力に自信を構築するのに役立ちます。

**プロジェクトを構築：**本を進めていく中で、興味のある個人プロジェクトに取り組むことを検討してください。これは研究からのデータ分析、コミュニティのための地図作成、または業務で遭遇した問題の解決などが考えられます。

**自分に忍耐強く：**プログラミングは特に学習中は苛立たいものです。エラーに遭遇し、デバッグに時間を費やし、時々行き詰まることを予期してください。これは正常で学習プロセスの一部です。必要な時は休憩を取り、専門知識は一貫した練習を通じて徐々に発達することを覚えておいてください。行き詰まった場合は、本の GitHub リポジトリで遠慮なく助けを求めてください。

**練習を続ける：**この本のスキルは維持し発展させるために定期的な練習が必要です。小さなものでも、地理空間プログラミングプロジェクトに取り組む時間を定期的に確保してください。

## この本で使用される慣例

この本では、内容をナビゲートし、コード例を理解するのに役立ついくつかの慣例を使用しています：

**コード書式：**すべての Python コードは、コードブロック内で等幅フォントで表示されます。通常のテキスト内にコードが表示される場合、このように 書式設定されます。ファイルとディレクトリ名も等幅フォントで書式設定されます。

**コード例：**ほとんどのコード例は完全で実行可能です。実証されている主要な概念と技術を説明するコメントが含まれています。付随するテキストでの参照のために行番号が含まれる場合があります。

```
# これはコードブロックの例です
import leafmap
m = leafmap.Map()
```

```
m.add_basemap("OpenTopoMap") # 地図にベースマップを追加
m
```

**コマンドライン指示:** コマンドラインまたはターミナルで入力するコマンドは \$ プロンプトで表示されます:

```
$ pip install leafmap
$ python script.py
```

## コード例のダウンロード

この本のすべてのコード例、データセット、補足資料は GitHub で自由に入手できます:

<https://github.com/giswqs/intro-gispro>

資料をダウンロードするには、以下の方法のいずれかを使用できます:

- **リポジトリをクローン** (Git がインストールされている場合):

```
$ git clone https://github.com/giswqs/intro-gispro.git
```

- **ZIP としてダウンロード** (Git を使用したくない場合):

- ▶ GitHub リポジトリページを訪問
- ▶ 緑色の **Code** ボタンをクリック
- ▶ **Download ZIP** を選択
- ▶ ファイルを希望の場所に解凍

- **個々のファイルを閲覧:** 特定の例のみが必要な場合は、GitHub インターフェースを通じてオンラインで閲覧

リポジトリは修正、改善、追加例で定期的に更新されます。定期的に確認するか、GitHub でリポジトリを **watch** して変更の通知を受け取ってください。

コードでエラーを見つけたり、改善の提案がある場合は、GitHub で **issue** を開くか **pull request** を提出してください。コミュニティの貢献は、このリソースをすべての人にとってより良いものにするのに役立ちます。

## ビデオチュートリアル

書面の内容を補完して、この本は主要な概念を解説し、追加の例を提供する包括的なビデオチュートリアルシリーズでサポートされています:

<https://tinyurl.com/intro-gispro-videos>

ビデオは書面の資料を補完するために、置き換えるためではなく設計されています。特に以下の場合に役立ちます:

- コードが書かれ実行されるのを見ることで利益を得る視覚的学習者
- 複数の説明を通じて複雑な概念を理解



- 開発ワークフローとベストプラクティスの学習
- 問題へのアプローチ方法とデバッグ問題の確認

プレイリストは本の構造に従って整理されています。本を進めながら順番に視聴することも、必要に応じて特定のトピックにジャンプすることもできます。

これらのビデオは、私がテネシー大学で [GIS プログラミング入門](#)<sup>1</sup> コースを教えていた 2024 年秋に作成されました。コースは終了しましたが、ビデオは関連性を保っており、本の参考として使用できます。将来的に追加のビデオが追加される予定です。

## お問い合わせ

読者からのフィードバック、質問、提案を歓迎します。あなたの意見は本の改善に役立ち、地理空間プログラミングコミュニティにとってより有用なものにします。

### 本に関する質問と議論について：

- GitHub Issues: <https://github.com/giswqs/intro-gispro/issues>
- GitHub Discussions: <https://github.com/giswqs/intro-gispro/discussions>

### 特に役立つフィードバックの種類：

- テキストやコードのエラーや不明確な説明
- 追加の例や使用例の提案
- 新しいトピックや章のアイデア
- 異なるオペレーティングシステムやライブラリバージョンでの互換性問題の報告
- 本の技術をどのように応用したかの成功事例

## 謝辞

この本は、多くの個人の貢献とサポート、そしてより広いオープンソース地理空間コミュニティなしには実現不可能でした。

**オープンソースコミュニティ：**この本は、Python 地理空間エコシステムを創造し維持してきた数え切れないオープンソース開発者の素晴らしい働きの上に構築されています。NumPy、Pandas、GeoPandas、Rasterio、Xarray、Rioxarray、Folium、ipyleaflet、MapLibre、GDAL、および地理空間プログラミングを利用しやすくする他の多くのライブラリの開発者と保守者に特別な感謝を。

**学生と同僚：**テネシー大学での地理空間プログラミングコースの学生からの質問、課題、洞察が、この本のアプローチと内容を形作りました。何が機能し何が機能しないかについての彼らのフィードバックは、真に学習者に役立つ教材を作成する上で貴重でした。

**研究協力者：**地理空間研究コミュニティの同僚と協力者は、本全体の実践的な例を形作る実世界の使用例、データセット、問題シナリオを提供してくれました。

**家族と友人：**技術書の執筆には多大な時間と集中力が必要です。このプロジェクトに捧げた多くの夜と週末を理解してくれた家族と友人の忍耐とサポートに感謝しています。

---

<sup>1</sup><https://geog-312.github.org>

**より広い GIS コミュニティ：** 地理空間分野は知識とツールの共有の基盤の上に構築されています。この本はその伝統の一部であり、地理空間プログラミング学習のために利用可能なリソースに貢献できることを光栄に思います。

この本は [MyST Markdown](https://mystmd.org)<sup>2</sup> を使って書かれ、[Typst](https://github.com/typst/typst)<sup>3</sup> と [min-book](https://github.com/mayconfmelo/min-book)<sup>4</sup> テンプレートを使ってコンパイルされました。Typst と MyST Markdown プロジェクトの開発者と保守者に感謝します。min-book テンプレートとこの本のためのテンプレートカスタマイズの支援について [Maycon F. Melo](https://github.com/mayconfmelo)<sup>5</sup> に特別な感謝を。

この本のエラーや省略は私の責任です。読者のフィードバックに基づいて問題に対処し、内容を改善することをお約束します。

## 著者について

呉秋生 (Dr. Qiusheng Wu) 博士は、テネシー大学ノックスビル校の地理学・持続可能性学部の准教授および大学院研究主任です。また、Amazon Scholar も務めています。Wu 博士の研究は地理空間データサイエンスとオープンソースソフトウェア開発に焦点を当てており、特に大規模地理空間データとクラウド計算を活用した環境変化、特に地表水と湿地浸水動態の研究に重点を置いています。彼は、高度な地理空間分析とインタラクティブ可視化をサポートする、[geemap](https://geemap.org)<sup>6</sup>、[leafmap](https://leafmap.org)<sup>7</sup>、[segment-geospatial](https://samgeo.github.io)<sup>8</sup>、[geoai](https://opengeoai.org)<sup>9</sup> を含む、広く使用されているいくつかのオープンソース Python パッケージの作成者です。彼のオープンソース作業は、GitHub の [Open Geospatial Solutions](https://github.com/opengeos)<sup>10</sup> で利用できます。

## ライセンスと著作権

この本はオープンサイエンスとオープン教育の原則を受け入れています。透明性、学習、再利用をサポートするため、この本の**コード例**は [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/) ライセンスの下でリリースされています。これは、適切なクレジットが与えられる限り、商用目的であってもコードをコピー、修正、配布することが自由であることを意味します。

コードの使用時は、本を引用するか GitHub リポジトリにリンクして帰属を明示してください：

Wu, Q. (2025). *Introduction to GIS Programming: A Practical Python Guide to Open Source Geospatial Tools*. <https://gispro.github.org>

コードは自由に利用できますが、この本の**テキスト**、**図**、**画像**は著者によって**著作権**が保護されており、明示的な許可なしに複製、再配布、または修正することはできません。これには、特に明記されていない限り、すべての書面の内容、カスタム図、埋め込み可視化が含まれます。

---

<sup>2</sup><https://mystmd.org>

<sup>3</sup><https://github.com/typst/typst>

<sup>4</sup><https://github.com/mayconfmelo/min-book>

<sup>5</sup><https://github.com/mayconfmelo>

<sup>6</sup><https://geemap.org>

<sup>7</sup><https://leafmap.org>

<sup>8</sup><https://samgeo.github.io>

<sup>9</sup><https://opengeoai.org>

<sup>10</sup><https://github.com/opengeos>

教育、プレゼンテーション、出版物など、本からコード以外の資料を再利用または適応させたい場合は、著者に連絡して許可を求めてください。

この二重ライセンスアプローチは、学習教材へのオープンアクセスと独創的な創作物の保護のバランスを取るのに役立ちます。これらの条件を尊重し、オープンソース地理空間コミュニティをサポートしていただき、ありがとうございます。

# GISプログラミング入門

## オープンソース地理空間ツールのための実践的Pythonガイド

空間データ分析の需要は、環境科学や都市計画から公衆衛生、ビジネスに至るまで、さまざまな分野で拡大し続けています。それに伴い、地理空間の複雑さに対応できる、アクセスしやすくスケーラブルなツールの必要性も高まっています。『PythonによるGISプログラミング入門』は、空間分析を習得するための実践的なガイドです。

本書は、学生、GIS専門家、データアナリスト、開発者など、あらゆるレベルの読者に向けて、Pythonを用いた地理空間プログラミングへの体系的かつ初心者優しい道筋を提供します。Pythonの基本から始め、GeoPandas、Rasterio、Leafmap、Geemap、GDALなどの強力なオープンソースライブラリを使って、高度な空間分析やインタラクティブな可視化へと段階的に進んでいきます。

本書の特長は、実例に基づいたアプローチにあります。実際のデータセットを扱い、現実の課題に取り組み、静的地図から動的なWebアプリケーションに至るまで、意味のあるアウトプットを作成していきます。各章は理論、注釈付きコード、実践課題、トラブルシューティングのヒントを組み合わせて、段階的にスキルを習得できるよう丁寧に構成されています。

### この本で学べること：

- conda、VS Code、Git、Docker、クラウドコンピューティングツールを使った開発環境の構築方法
- 変数やデータ構造、ファイル入出力、Pandasなどの基本的なPythonプログラミングスキル
- ベクター／ラスターのデータ処理、インタラクティブマッピング、3D可視化、地理空間解析
- Google Earth Engineを活用したクラウドベースの地理空間コンピューティングと、ハイパースペクトルデータや空間SQLなどの応用トピック

呉 秋生 (Qiusheng Wu) 博士は、オープンな地理空間データサイエンス分野における第一人者であり、研究者・教育者として活躍しています。彼は、geemap、leafmap、segment-geospatial、geoai など、広く利用されている複数のオープンソースPythonパッケージの開発者です。オンラインでの情報はこちら：

- <https://x.com/giswqs>
- <https://linkedin.com/in/giswqs>
- <https://youtube.com/@giswqs>
- <https://github.com/giswqs>



QRコードをスキャンして書籍のウェブサイト

(<https://gispro.gishub.org>) にアクセスし、コード例をダウンロードしてください。