

ArcGIS

ArcPy スタートアップガイド

目次

はじめに.....	5
第 1 章 イントロダクション	6
ArcPy とは.....	7
ArcPy の利点.....	7
Python とは.....	8
演習 1A: 環境設定	9
ステップ 1: ArcGIS Desktop のインストール	9
ステップ 2: PyScripter のインストール.....	9
ステップ 3: PyScripter の環境設定.....	10
ステップ 4: Python アドイン ウィザードのダウンロード	13
ステップ 5: ArcGIS Pro のインストール.....	14
演習 1B: データのダウンロードと事前処理	15
ステップ 1: データ保存先フォルダーの作成	15
ステップ 2: 全国市区町村界データのダウンロード	16
ステップ 3: 全国市区町村界から新潟県のフィーチャを エクスポート.....	19
ステップ 4: 長岡市データのダウンロードと人口データのテーブル結合	21
ステップ 5: 警察署のデータをダウンロードし、投影法を定義	29
第 2 章 ArcPy の基本操作	33
ArcPy の操作方法	34
Python ウィンドウ	34
フィールド演算.....	35
統合開発環境 (IDE) を使用した ArcPy の実行	36
演習 2A: [Python] ウィンドウによる操作	38
ステップ 1: [ディゾルブ] ツールの実行	38
演習 2B: フィールド演算による操作.....	42
ステップ 1: 単純なフィールド演算の実行.....	42
ステップ 2: 高度なフィールド演算の実行.....	47
演習 2C: 統合開発環境 (IDE) による操作	51
ステップ 1: Python インタプリタによる操作.....	51
ステップ 2: Python モジュールによるプログラムの実行	53
第 3 章 Python を使用したツールの作成	57
スクリプト ツールとは.....	58
演習 3: スクリプト ツールの作成	59
ステップ 1: スクリプト ファイルの作成	59
ステップ 2: スクリプト ツールの作成	61
ステップ 3: スクリプト ツールの実行	65
第 4 章 Python アドインの作成	68

アドインの作成方法	69
アドインとは.....	69
Python で作成できるアドインの種類	69
演習 4A: ボタンの作成	70
ステップ 1: プロジェクトの作成	70
ステップ 2: コードの記述	72
ステップ 3: 動作のテスト	73
演習 4B: ツールの作成	76
ステップ 1: プロジェクトの作成	76
ステップ 2: コードの記述	78
ステップ 3: 動作のテスト	79
第 5 章 ArcGIS Pro での操作	82
ArcGIS Pro での Python の利用	83
Python ウィンドウ	83
外部アプリケーションから Arcpy を実行.....	83
arcpy の機能に対する変更.....	83
Python 3.x へのアップグレード	84
サポートされていないデータ形式.....	84
演習 5: スクリプト ツールのチェック	85
ステップ 1: プロジェクトの作成	85
ステップ 2: チェック ツールの実行	86
ステップ 3: スクリプト ツールの実行	90
第 6 章 補足資料.....	94
参考書籍	95
Web サイト.....	95
サービス	95

はじめに

このガイドは「ArcPy」を使用して ArcGIS を操作する方法を学ぶためのチュートリアルです。ArcPy を使用するとどんな利点があるのか、どのような作業の効率化が図れるのかを演習形式で操作しながら理解していただけます。

利用製品

このガイドでは、以下の製品を前提としています。

- ✓ ArcGIS Desktop 10.7 (Basic、Standard、Advanced のいずれでも動作)
- ✓ ArcGIS Pro 2.3 (Basic、Standard、Advanced のいずれでも動作)

前提知識

このガイドは、ArcGIS Desktop、ArcGIS Pro および Python 言語をご存じでない方でも操作できるように記載しております。ただし、ArcMap の基本操作と Python 言語の基本構文を理解していることが望ましいです。

アイコンの説明



演習時間: 演習を終了するために必要とされる標準時間の概算です。



ノート: 特定のトピック、手順に関する追加の情報、例外事項や特記事項を示します。



ティップス: 概念の理解や手順を実行するための簡単なヘルプです。



外部リソース: トピックに関する参考資料などを示します。



ベスト プラクティス: 目的や優先事項を達成し、かつ時間を短縮するための一般的、あるいは専門的なガイドラインです。



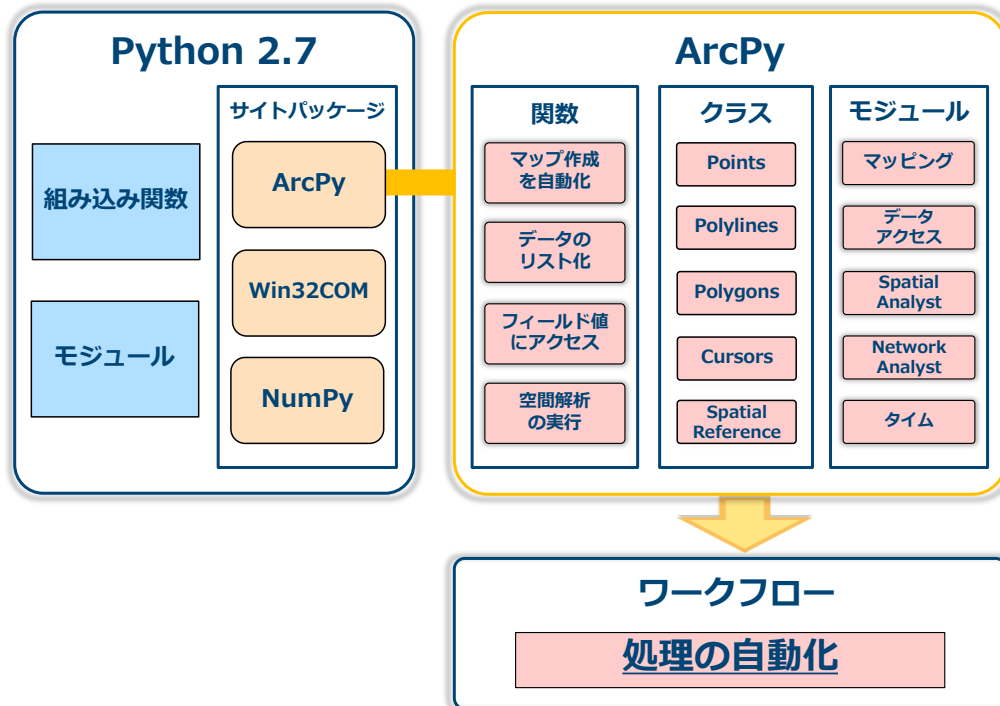
警告: 問題となりうる箇所や、避けるべき行動を示します。



第 1 章 イントロダクション

ArcPy とは

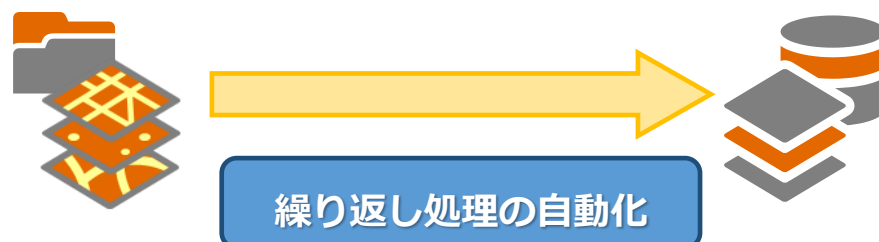
ArcPy とは、プログラム言語である Python を使用して ArcGIS を操作するために Esri が開発した API です。「ArcPy サイト パッケージ」を使用することで、ArcGIS のマップ作成やフィールド演算、ジオプロセッシング ツールといった、ArcGIS 独自の機能が Python 言語で実行できます。



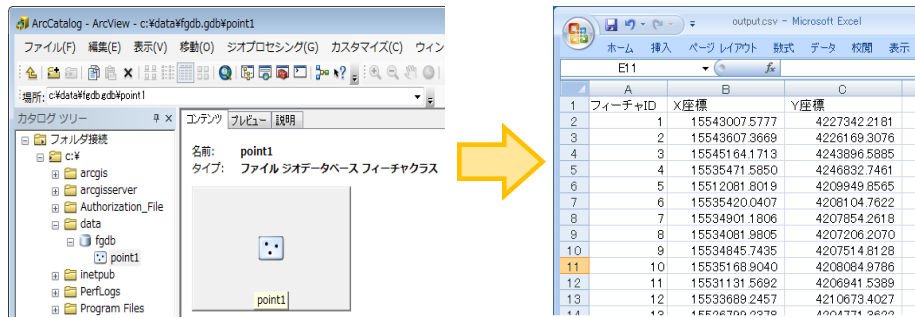
ArcPy の利点

ArcPy を使用することで ArcGIS 上で GIS データを操作する便利なプログラムが作成できます。

- ✓ 繰り返し作業を自動化させ、作業効率を上げる
 - 例) フォルダー内にあるシェープファイルを、繰り返し処理を自動化させて、すべてジオデータベースに変換



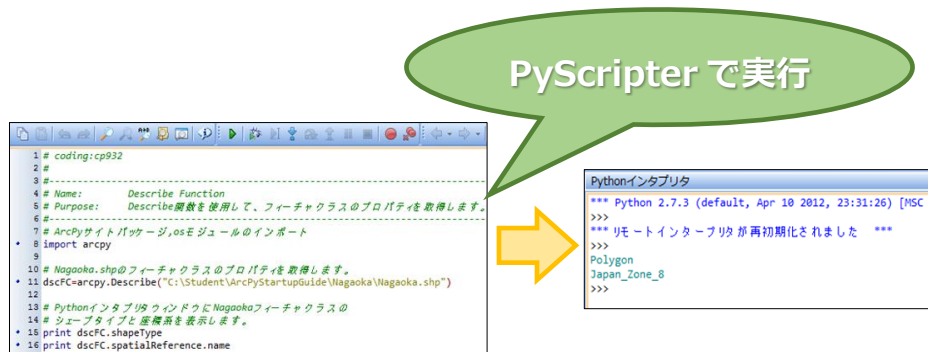
- 例) ジオデータベースのポイント フィーチャクラスに含まれるポイントの ID と座標値を CSV ファイルに出力する繰り返し作業の自動化



フィーチャクラスから CSV への出力作業を自動

✓ ArcMap を起動せずに作業を行う

- 例) PyScripter からポリゴン フィーチャクラスのプロパティ情報の取得



シェープタイプと座標系のプロパティ情報を取得

Python とは

Python とはプログラミング言語の一つです。オープンソースで提供され、ソースコードをコンピュータ上で実行可能な形式であるオブジェクト コードに変換する、あるいはコンパイルを必要としないスクリプト言語です。可読性が高く、以前書いたコードの編集が行いやすいという利点があります。Python は Google や Facebook といった企業でも使用されています。



Python についての詳しい学習は、第 6 章の補足資料をご参考ください。

演習 1A: 環境設定

ArcPy を使用するには、事前に実行環境である ArcGIS Desktop がインストールされている必要があります。また、このガイドでは Python 言語を記述するための統合開発環境として PyScripter を使用します。事前準備として、ArcGIS Desktop と PyScripter のインストール、さらに、第 4 章で学習する Python アドインの作成に必要な Python アドイン ウィザードをダウンロードします。そして、第 6 章で学習する ArcGIS Pro での ArcPy の利用方法に必要な ArcGIS Pro をインストールします。



演習時間 : 15 分

ステップ 1: ArcGIS Desktop のインストール

ArcPy を使用するには、事前に実行環境である ArcGIS Desktop がインストールされている必要があります。ArcGIS Desktop のインストールを行きましょう。

ArcGIS Desktop 10.7 のインストール要件については、ESRI ジャパンサイト内の製品の動作環境ページを参照してください。

ArcMap 動作環境 (10.7)

<https://www.esri.jp/products/arcgis-desktop/environments/arcmap/>

ソフトウェアのダウンロードとインストール方法は以下の URL を参照してください。

ArcGIS Desktop 10.7

https://esri-esri-support.custhelp.com/app/answers/detail/a_id/7579



ArcGIS Desktop 10.7 をインストールするには、Esri 製品サポート サイトにアクセスし、ログインする必要があります。アカウントをお持ちでない場合は、Esri 製品サポート トップページを参照してください。

ステップ 2: PyScripter のインストール

ArcGIS Desktop をインストールするコンピュータに Python 2.7.15 と Numerical Python (NumPy) 1.9.3 が存在しない場合、ArcGIS Desktop のインストーラーにより、Python 2.7.15 と Numerical Python 1.9.3 がインストールされます。

- 以下のページへアクセスし、「PyScripter-v3.6.0-Setup.exe」ファイルをダウンロードし、任意の場所に保存します。

PyScripter

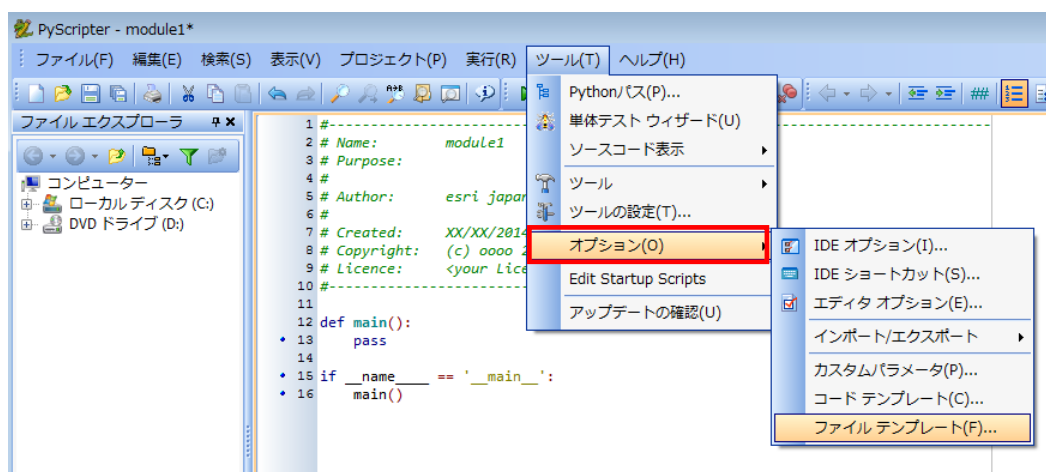
<https://sourceforge.net/projects/pyscripter/>

- ダウンロードした exe ファイルをダブルクリックします。
- [Setup-PyScripter] ダイアログが表示されるので、[Next] をクリックします。
- 再度 [Next] をクリックし PyScripter のインストール場所を設定後、[Next] をクリックします。
- インストール先に作成される PyScripter のフォルダー名を設定し、[Next] をクリックします。
- Windows のデスクトップ画面や、タスクバーにショートカット アイコンを作成するかを設定し [Next] をクリックします。
- インストール内容を確認し、[Install] をクリックします。
- 正常にインストールが完了したら、[Information] という画面に変わりますので、[Next] をクリックします。
- [Finish] をクリックし、インストールを終了します。

ステップ 3: PyScripter の環境設定

PyScripter のテンプレート フォーマットを変更します。

- PyScripter を起動します (デスクトップのショートカットアイコン、または Windows の [スタート] プログラム → [PyScripter] → [PyScripter for Python 2.7])。
- PyScripter の [ツール] メニューから [オプション] → [ファイル テンプレート] を選択します。

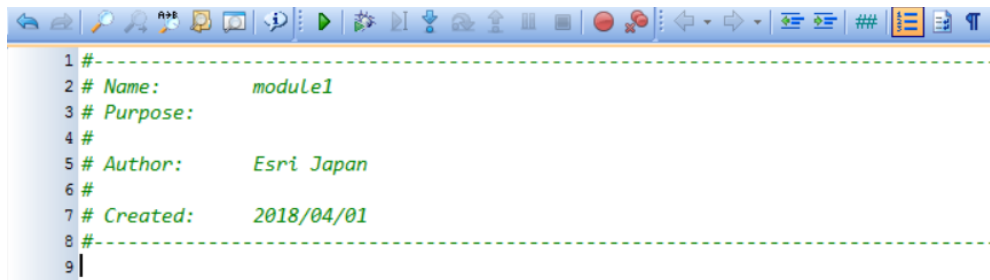


- [ファイル テンプレート] ダイアログの上部にあるウィンドウから [Python スクリプト] を選択します。
- [ファイル テンプレート] ダイアログの [テンプレート] 部分に記載されているコードを

以下のように変更します。

```
# coding:cp932
#
#-----
# Name:      ${ActiveDoc-Name}
# Purpose:
#
# Author:    ${UserName}
#
# Created:    ${DateTime-'YYYY/MM/DD'-DateFormat}
#-----
#
```

- [ファイル テンプレート] ダイアログにある [更新] ボタンをクリックし、[OK] ボタンをクリックします。
- テンプレートの更新を確認するために、[ファイル] メニューの [新規作成] → [新規 Python モジュール] を選択します。
- 変更したテンプレートが適用されたことを確認します。



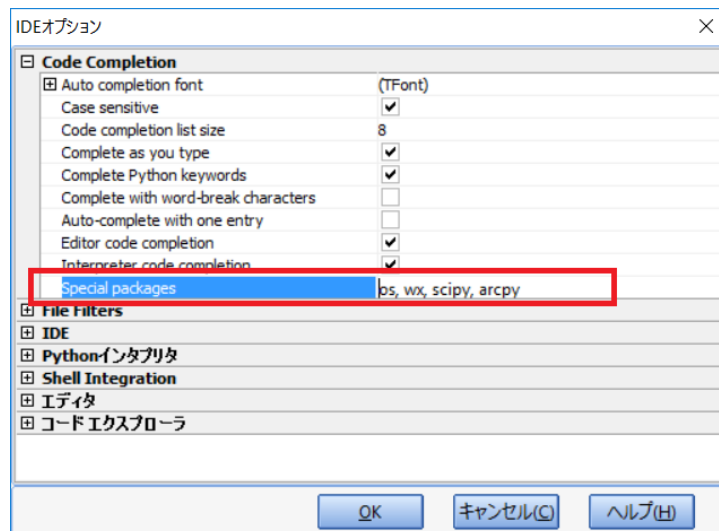
```
1 #-----
2 # Name:      module1
3 # Purpose:
4 #
5 # Author:    Esri Japan
6 #
7 # Created:    2018/04/01
8 #-----
9 |
```

次に、PyScripter のオプション設定を行います。PyScripter のデフォルト設定では、PyScripter を一度終了し、再度起動した場合、PyScripter を閉じる前に使用していたスクリプト ファイルが表示されるようになっています。このコースでは、PyScripter を起動するたびに、新しいスクリプト ファイルが開くように設定します。

- PyScripter の [ツール] メニューから [オプション] → [IDE オプション] を選択します。
- [IDE オプション] ダイアログ下部の [エディタ] 項目にある [Restore open files] のチェックを外します。

続いて、PyScripter 上で ArcPy サイト パッケージのオート コンプリートを有効にする設定を行います。

- [IDE オプション] ダイアログ上部の [Code Completion] 項目にある [Special packages] に「arcpy」を追加します。

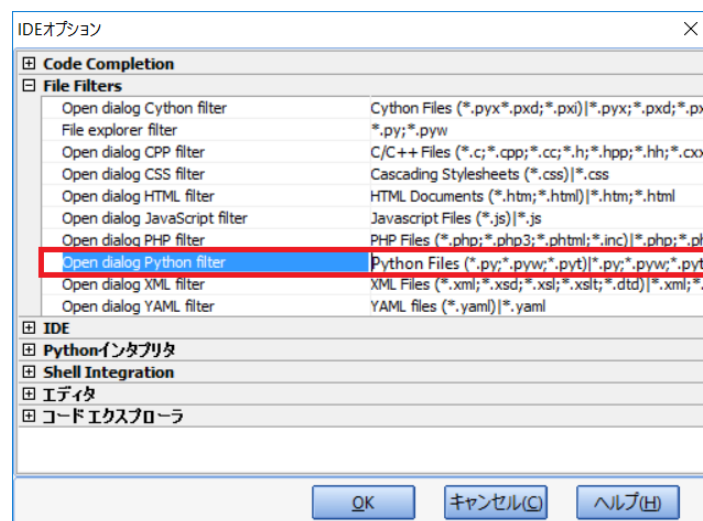


[Spatial packages] に「arcpy」を入力すると、初回インポート時に ArcPy サイト パッケージのすべてのモジュールを読み込むため、インポートに時間がかかる可能性があります。

また、PyScripterが認識する Python スクリプト ファイルは、デフォルト設定では「.py」ファイル形式のみとなっていますので、ArcMap の Python ツールボックスで使用する「.pyt」ファイル形式を PyScripter で読み込めるよう関連付けます。

- [IDE オプション] ダイアログの [File Filters] 項目にある [Open dialog Python filter] に「*.pyt」を 2 箇所追加します。

Python Files (*.py;*.pyw;*.pyt) | *.py;*.pyw;*.pyt



次に、[Pythonインタプリタ] ウィンドウで使用する文字コードを変更します。

- [IDE オプション] ダイアログの [Pythonインタプリタ] 項目にある [UTF8 in interactive interpreter] のチェックを外します。これにより、Shift-JIS に指定されます。

[IDE オプション] の [OK] をクリックします。

最後に [Pythonインタプリタ] ウィンドウに行番号を表示します。

- [ツール] メニュー → [オプション] → [エディタ オプション] をクリックします。
- [エディタ オプション] ダイアログの [表示] タブで次の 2 ヶ所を設定し、[OK] をクリックします。
 - ・ [ガター (Gutter)] → [行番号の表示] にチェックを入れます。
 - ・ [右のエッジ] → [エッジのカラム] を 0 に変更します。

ステップ 4: Python アドイン ウィザードのダウンロード

Python アドイン ウィザードを使用することで、容易に ArcMap のカスタム UI (ボタンやツール) などを作成することができます。このステップでは、Python アドイン ウィザードをダウンロードします。

- 以下のページにアクセスして [開く] - [ダウンロード] を選択し、「addin_assistant.zip」ファイルを任意の場所に保存してください。

ArcGIS - Python Add-In Wizard

<http://www.arcgis.com/home/item.html?id=5f3aefe77f6b4f61ad3e4c62f30bff3b>

Python Add-In Wizard

概要



Simplify the development of ArcGIS Desktop add-ins with this Python Add-In Wizard

📄 Desktop Application Template 作成者 [teampython](#)

作成日時: 2012年1月4日 更新日: 2013年6月13日

ビュー数: 55,223

詳細

ダウンロード

- ダウンロードした「addin_assistant.zip」ファイルを解凍します。
- ファイルの解凍により作成された「addin_assistant」フォルダー、または「addin_assistant」フォルダー内にある「bin」フォルダーを任意のフォルダーにコピーします。
- 「bin」フォルダー内の「addin_assistant.exe」ファイルのショートカットをデスクトップなどに配置します。

addin_assistant.exe を実行すると、Python アドイン ウィザードが起動できます。

ステップ 5: ArcGIS Pro のインストール

第 5 章を行う場合は ArcGIS Pro をインストールします。

ArcGIS Pro のシステム要件については、下記ヘルプ ページを参照してください。

ArcGIS Pro 2.3 のシステム要件

<http://pro.arcgis.com/ja/pro-app/get-started/arcgis-pro-system-requirements.htm>

ソフトウェアのダウンロードとインストール方法は以下の URL を参照してください。

ArcGIS Pro 2.3

https://esri-esri-support.custhelp.com/app/answers/detail/a_id/7681



ArcGIS Pro をインストールするには、Esri 製品サポート サイトにアクセスし、ログインする必要があります。アカウントをお持ちでない場合は、Esri 製品サポート トップページを参照してください。

以上で、事前の環境設定は終了です。

演習 1B: データのダウンロードと事前処理

この演習では、このガイドの演習で使用するデータをダウンロードし、実行可能な形式に編集します。作業ごとに、ArcMap を新規に起動してデータの編集を行ってください。



演習時間 : 20 分

ステップ 1: データ保存先フォルダーの作成

このステップでは、演習で使用するデータの保存先フォルダーを作成します。

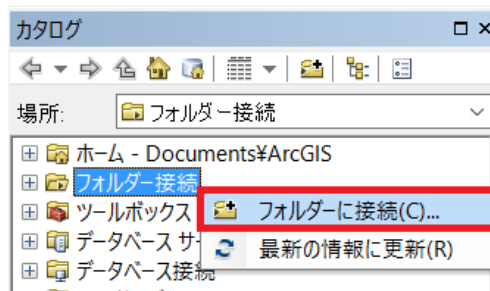
- 「C:¥」に「ArcPyStartupGuide」フォルダーを作成します。
- さらに「C:¥ArcPyStartupGuide」フォルダー内に、以下のフォルダーを作成します。
 - Downloads
 - Geodatabases
 - MapDocuments
 - Projects
 - PythonAddins
 - Scripts

この演習でダウンロードしたデータは「C:¥ArcPyStartupGuide¥Downloads」フォルダーに保存します。また、このガイドではダウンロードしたデータをファイル ジオデータベースのフィーチャクラスに変換して演習を行います。変換したデータを格納するためのファイル ジオデータベースを構築します。

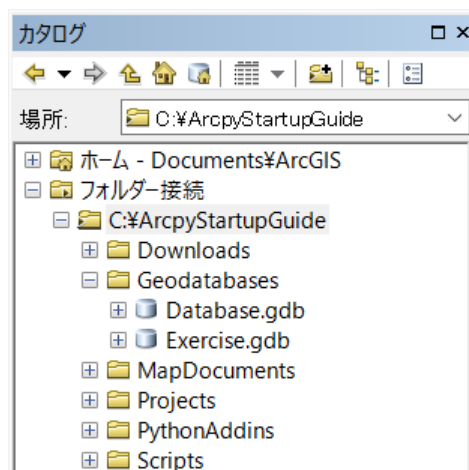
- ArcMap を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcMap 10.7])。
- [ArcMap - はじめに] ダイアログが表示されたら [キャンセル] をクリックします。
- [ツール] ツールバーの [カタログ] ボタンをクリックし、[カタログ] ウィンドウを開きます。



- [カタログ] ウィンドウの [フォルダー接続] を右クリックし、[フォルダーに接続] をクリックして以下のパスを参照して [OK] をクリックします。
 - C:¥ArcPyStartupGuide



- 「C:\ArcPyStartupGuide\Geodatabases」フォルダー上で右クリックし、[新規作成] → [ファイル ジオデータベース] をクリックします。
- 作成された「New File Geodatabase.gdb」を左クリックし、名前を「Database.gdb」に変更します。
- 同様に「C:\ArcPyStartupGuide\Geodatabases」フォルダー内に新規ファイル ジオデータベースを作成し、名前を「Exercise.gdb」とします。
- [カタログ] ウィンドウを表示して、下図の状態であることを確認してください。



ステップ 2: 全国市区町村界データのダウンロード

Web サイトから無料で提供している、全国市区町村界データをダウンロードします。このデータは、演習 2A と演習 2B のステップ 1 で使用します。

- Web ブラウザー (Internet Explorer) を開き、以下の URL にアクセスします。

ESRI ジャパン 全国市区町村界データ
<http://www.esri.jp/products/japan-shp/>

esri ジャパン

全国市区町村界データ

全国市区町村界データは、ArcGIS の利便性向上のために、ESRIジャパン株式会社がサンプルデータとして無償で配布しているものです。

使用規定で許諾された範囲内でのみご使用いただけます。
(全国市区町村界データの使用規定は[こちら](#) (PDF 80.5KB))

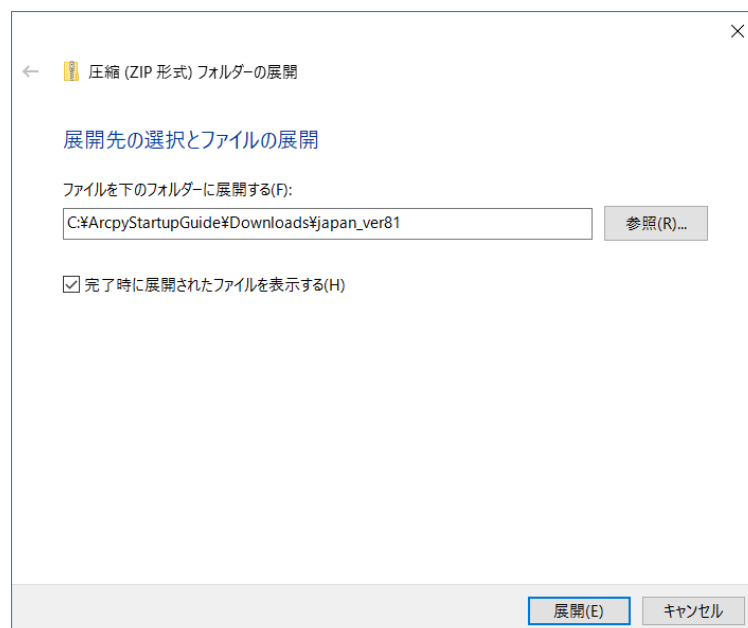
ファイルのダウンロード

データ仕様

フォーマット	シェープファイル
図形タイプ	ポリゴン
座標系	地理座標系 日本測地系 2000 (緯度経度 JGD 2000)

属性情報

- [ファイルのダウンロード] をクリックし、内容を確認してから [同意する] にチェックを入れ、[全国市区町村界データのダウンロード] をクリックします。
 - [名前を付けて保存] を選択し、「C:¥ArcPyStartupGuide¥Downloads」フォルダーに、「japan_ver81.zip」と名前を付けて保存してください。
- ダウンロードしたファイルを、同じパス内に解凍します。
- 「C:¥ArcPyStartupGuide¥Downloads」フォルダーへ移動し、「japan_ver81.zip」上で右クリックして [すべて展開] を選択します。
 - [圧縮 (ZIP 形式) フォルダーの展開] ダイアログが表示されたら、[完了時に展開されたファイルを表示する] にチェックを入れ、[展開] をクリックします。



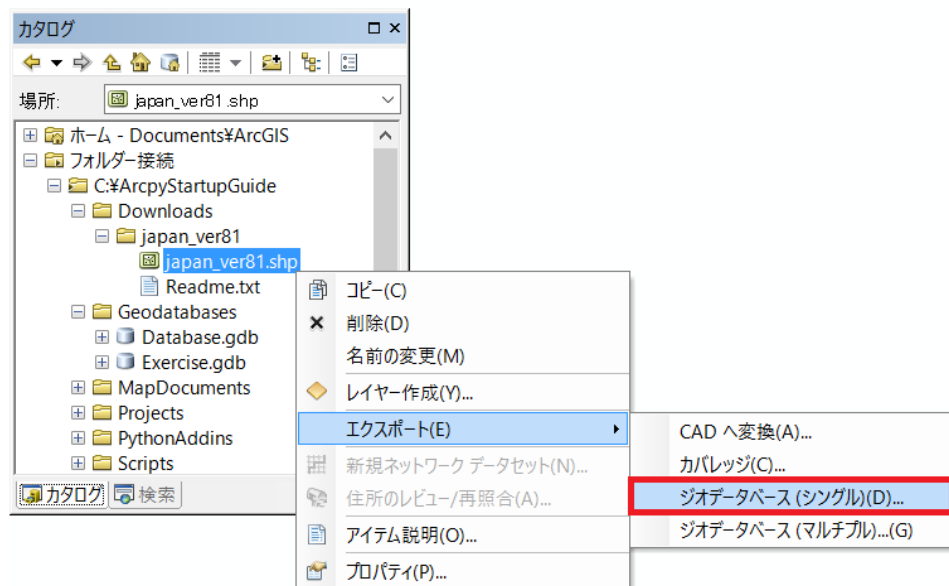
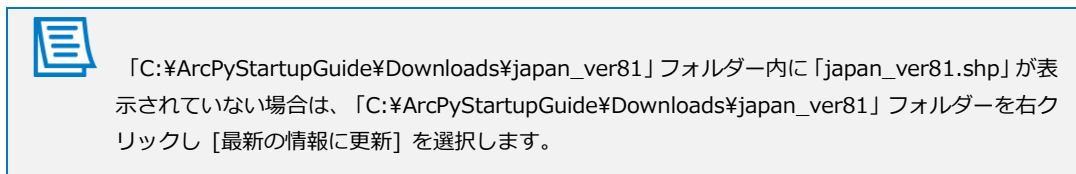
「japan_ver81」フォルダーが「C:¥ArcPyStartupGuide¥Downloads」フォルダー内に解凍されたことを確認します。フォルダーを開くと「japan_ver81.shp」というシェープファイルが格納されていることが確認できます。

次にダウンロードした「japan_ver81.shp」を、ファイル ジオデータベースのフィーチャクラスに変換します。

□ ArcMap が起動していない場合は、以下の操作を行います

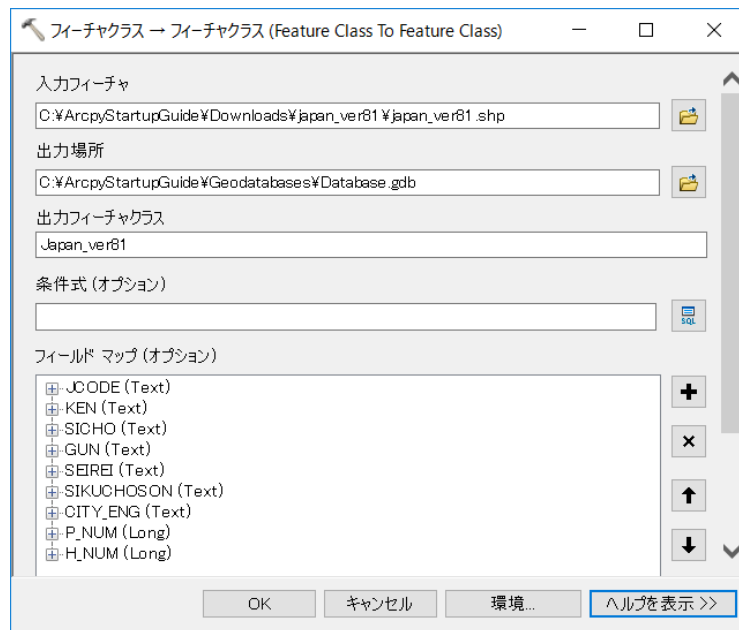
- ArcMap の起動
- [カタログ] ウィンドウの表示

□ [カタログ] ウィンドウで「C:¥ArcPyStartupGuide¥Downloads¥japan_ver81」フォルダーへ移動し、[japan_ver81.shp] 上で右クリックし、[エクスポート] → [ジオデータベース (シングル)] を選択します。

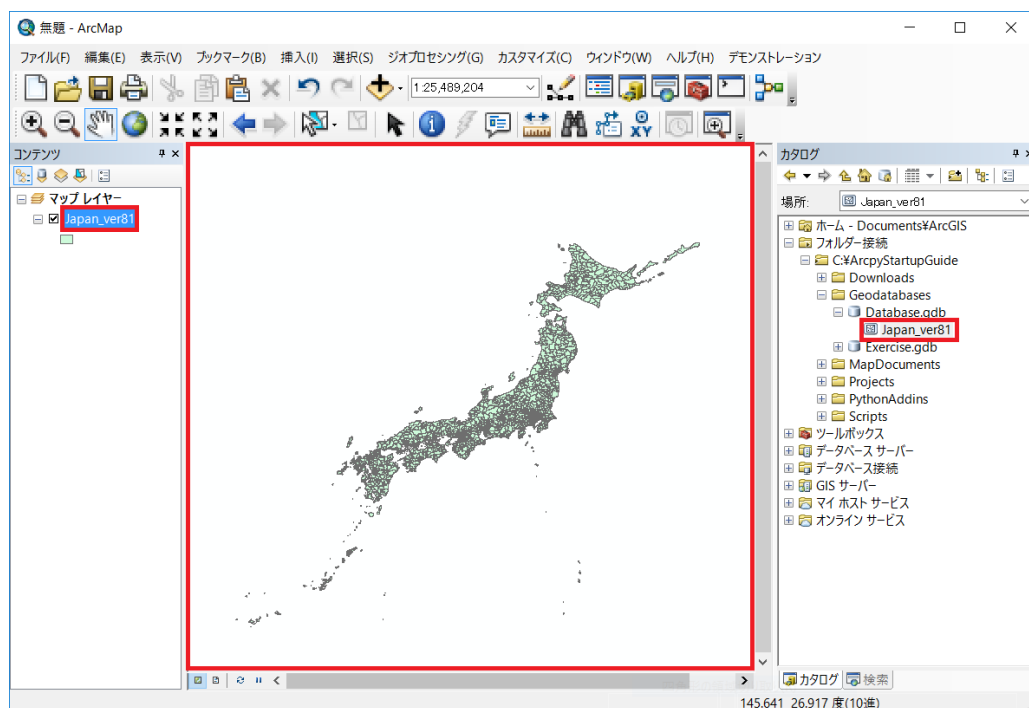


□ 表示された [フィーチャクラス → フィーチャクラス] ダイアログで以下の設定を行い、[OK] をクリックします。

- 入力フィーチャ : 「C:¥ArcPyStartupGuide¥Downloads¥japan_ver81¥japan_ver81.shp」
(既定値)
- 出力場所 : 「C:¥ArcPyStartupGuide¥Geodatabases¥Database.gdb」を選択
- 出力フィーチャクラス : 「Japan_ver81」と入力



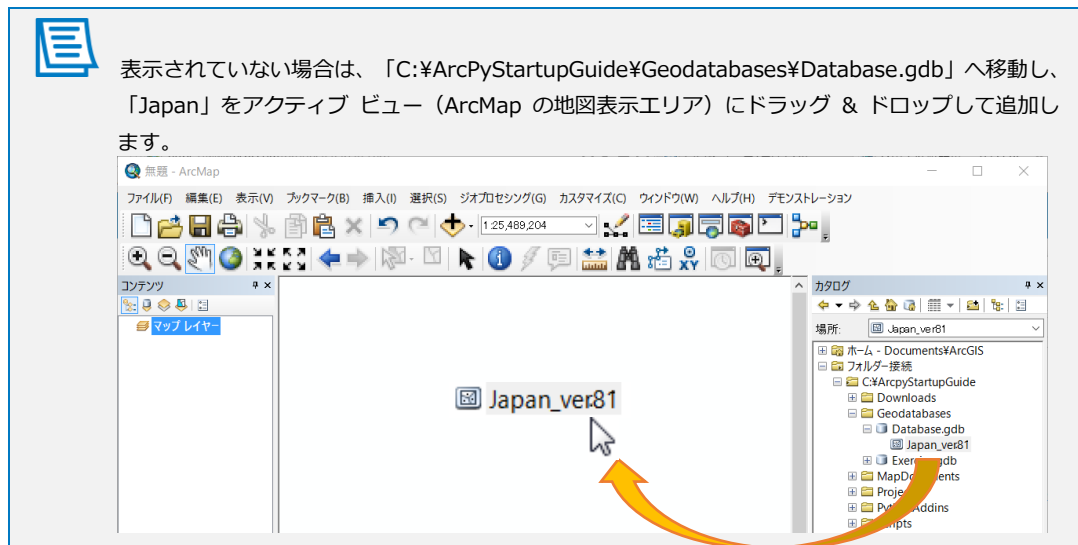
操作が完了すると、「Database.gdb」内に「Japan_ver81」フィーチャクラスが作成されます。また、マップ上および [コンテンツ] ウィンドウに「Japan_ver81」データがレイヤーとして追加されます。



ステップ 3: 全国市区町村界から新潟県のフィーチャを エクスポート

全国市区町村界レイヤーから「新潟県」のみを取り出し、新規に「Niigata」フィーチャクラスを作成します。このデータは、演習 2C で使用します。

- ArcMap のマップ上に「Japan_ver81」レイヤーが表示されていることを確認します。



- ArcMap の [選択] メニューから [属性検索] を選択し、以下の設定を行い [OK] をクリックします。

- レイヤー : 「Japan_ver81」を選択
- 選択方法 : [新規選択セットの作成] を選択
- SELECT * FROM Japan_ver81 WHERE : 以下の条件式を記述

KEN = '新潟県'



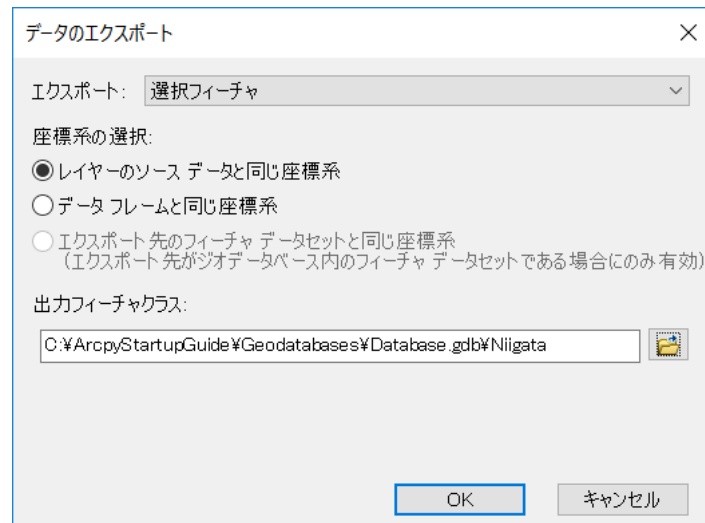
条件式の文字列は、[選択方法] の下のブロック内にある [KEN] をダブルクリックし、[個別値の取得] をクリックして表示された「新潟県」をダブルクリックすると入力できます。

「Japan_ver81」レイヤーの [KEN] フィールド内の「新潟県」のみが選択された状態になります。次に選択フィーチャをエクスポートして新しいフィーチャクラスを作成します。

□ [コンテンツ] ウィンドウ の「Japan_ver81」レイヤー上で右クリックし、[データ] → [データのエクスポート] を選択し、以下のパラメーターを設定して [OK] をクリックします。

- エクスポート : [選択フィーチャ] を選択
- 座標系の選択 : [レイヤーのソース データと同じ座標系] にチェック
- 出力フィーチャクラス :

「C:\¥ArcPyStartupGuide¥Geodatabases¥Database.gdb¥Niigata」を入力



□ [マップにレイヤーとしてエクスポート データを追加しますか?] と聞かれるので [はい] をクリックします。

操作が完了すると、「Database.gdb」内に「Niigata」フィーチャクラスが作成されます。また、マップ上および [コンテンツ] ウィンドウに「Niigata」データがレイヤーとして追加されます。

ステップ 4: 長岡市データのダウンロードと人口データのテーブル結合

e-Stat サイトから長岡市の町村界データと、男女別の人口データをダウンロードします。人口データを町村界データにテーブル結合し、男女別人口を格納するデータを新規にファイル ジオデータベースのフィーチャクラスとして出力します。ここで作成したデータは、演習 2 B で使用します。

□ Web ブラウザー (Internet Explorer) を開き、以下の URL にアクセスします。

e-Stat 政府統計の総合窓口
<http://www.e-stat.go.jp/>

まずは、男女別の人口データをダウンロードしましょう。

- 男女別の人口データをダウンロード [地図で見る] → [統計データダウンロード] を選択します。



- [データ ダウンロード] ページで、[国勢調査] → [2015年] → [小地域 (町丁・字等別)] → [男女別人口総数及び世帯総数] の順にクリックします。

地図で見る統計(統計GIS)
データダウンロード

地域メッシュ統計とは

+ 国勢調査	公開 (更新) 日	定義書
- 2015年		
- 小地域 (町丁・字等別)	2017-12-25	
男女別人口総数及び世帯総数	2017-06-29	定義書
年齢 (5歳階級、4区分) 別、男女別人口	2017-06-29	定義書

- [15 新潟県] を探し、[CSV] をクリックします。

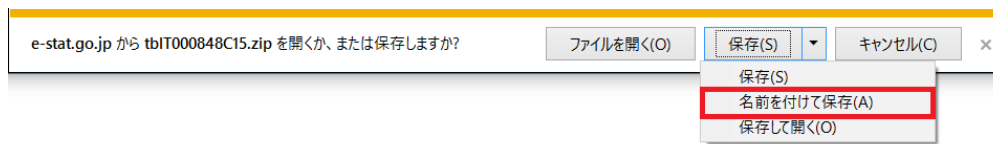
地図で見る統計(統計GIS)
データダウンロード

定義書

1/3ページ

統計表*	地域*	公開 (更新) 日*	形式
男女別人口総数及び世帯総数	01 北海道	2017-06-29	CSV
	⋮		
男女別人口総数及び世帯総数	15 新潟県	2017-06-29	CSV

- [e-stat.go.jp から tbIT000848C15.zip を開くか、または保存しますか?] というダイアログが表示されたら、[保存] の ドロップダウン リストから [名前を付けて保存] を選択します。



- 「C:¥ArcPyStartupGuide¥Downloads」フォルダーに「Population.zip」と名前を付けて保存します。

続いて、境界データのダウンロードを行いましょう。

- [地図で見る] → [境界データダウンロード]を選択します。



- 境界一覧から [小地域] を選択します。
- 次のページでは、[国勢調査] → [2015年] → [小地域 (町丁・字等別)] の順にクリックします。
- データ形式の一覧から [世界測地系平面直角座標系・Shape形式] を選びます。
- 都道府県の一覧から [15 新潟県] をクリックします。
- 新潟県の地域の一覧から [15020 長岡市] を探し、[世界測地系平面直角座標系・Shape形式] をクリックします。

地域*	公開(更新)日*	形式
15000 新潟県全域	2017-06-29	世界測地系平面直角座標系・Shape形式
⋮		
15202 長岡市	2017-06-29	世界測地系平面直角座標系・Shape形式

- [e-stat.go.jp から A002005212015XYSWC15202.zip を開くか、または保存しますか?] というダイアログが表示されたら、[保存] の ドロップダウン リストから [名前を付けて保存] を選択します。
- 「C:¥ArcPyStartupGuide¥Downloads」フォルダーに「Nagaoka.zip」と名前を付けて保存します。

ダウンロードした「Population.zip」と「Nagaoka.zip」を、同じパス内に解凍します。

- 「C:¥ArcPyStartupGuide¥Downloads」フォルダーへ移動し、「Population.zip」ファイルを右クリックして [すべて展開] をクリックします。
- [圧縮 (ZIP 形式) フォルダーの展開] ダイアログが表示されたら、[完了時に展開されたファイルを表示する] にチェックを入れ、[展開] をクリックします。

フォルダーが「C:¥ArcPyStartupGuide¥Downloads」フォルダーに解凍されたことを確認します。フォルダーを開くと「tbIT000848C15.txt」というテキストファイルが格納されていることが確認できます。同様に「Nagaoka」フォルダーも解凍します。

- 「C:¥ArcPyStartupGuide¥Downloads」フォルダーへ移動し、「Nagaoka.zip」ファイルを右クリックして [すべて展開] をクリックします。
- [圧縮 (ZIP 形式) フォルダーの展開] ダイアログが表示されたら、[完了時に展開されたファイルを表示する] にチェックを入れ、[展開] をクリックします。
- フォルダーが「C:¥ArcPyStartupGuide¥Downloads」フォルダーに解凍されたことを確認します。フォルダーを開くと「h27ka15202.shp」というシェープファイルが格納されていることが確認できます。

次に「男女別人口データ」を持つ「長岡市の町村界データ」を作成するために、「tbIT000848C15.txt」を「h27ka15202.shp」にテーブル結合します。

- ArcMap のアクティブ ビューに追加しているレイヤーをすべて非表示にします。
- ArcMap の [カタログ] ウィンドウで「C:¥ArcPyStartupGuide¥Downloads¥Nagaoka」フォルダーへ移動し、「h27ka15202.shp」をドラッグ & ドロップでアクティブ ビューに追加します。
- [カタログ] ウィンドウで「C:¥ArcPyStartupGuide¥Downloads¥Population」フォルダーへ移動し、「tbIT000848C15.txt」をドラッグ & ドロップでアクティブ ビューに追加します。



「C:\ArcPyStartupGuide\Downloads」フォルダー内に「Nagaoka」と「Population」フォルダーが表示されていない場合は、「C:\ArcPyStartupGuide\Downloads」フォルダーを右クリックし「最新の情報に更新」を選択します。

- [コンテンツ] ウィンドウで「h27ka15202」レイヤーを右クリックし、[属性の結合とリレート] → [結合] を選択し以下の設定を行い、[OK] をクリックします。

- このレイヤーへの結合の対象は? : [テーブルの属性を結合] (既定値)
- 結合に利用する値を持つフィールド : [MOJI] を選択
- 結合対象レイヤーまたはテーブル : [tblT000848C15.txt] を選択
- 結合のマッチングに利用するフィールド : [NAME] を選択
- 結合オプション : [すべてのレコードを保持] にチェック

結合

結合により、このレイヤーの属性テーブルにデータを追加することができます。

このレイヤーへの結合の対象は? (W)

テーブルの属性を結合

1. 結合に利用する値を持つフィールド(O):

MOJI

2. 結合対象レイヤーまたはテーブル(T):

tblT000848C15.txt

☒ このリストにレイヤーの属性テーブルも表示(S)

3. 結合のマッチングに利用するフィールド(F):

NAME

結合オプション

☒ 全てのレコードを保持(K)

結合対象テーブルのすべてのレコードが出力テーブルに表示されます。一致しないレコードには、結合対象テーブルによって追加されたすべてのフィールドに NULL 値が格納されます。

☐ 一致するレコードのみを保持(M)

元のテーブルのデータと結合対象テーブルのレコードが一致しない場合、そのレコードは出力テーブルから削除されます。

結合の整合チェック(V)

[データの結合について](#)

OK キャンセル

- [コンテンツ] ウィンドウで「h27ka15202」レイヤーを右クリックし、[属性テーブルを開く] をクリックしてデータが結合されているか確認します。

「T000848002」フィールドが男性、「T000848003」フィールドが女性の人口数データです。

上記の 2 つのフィールドは、[Text] 型のフィールドで保存されており、フィールド演算で計算ができないため、両方とも数値型のフィールドに変更します。

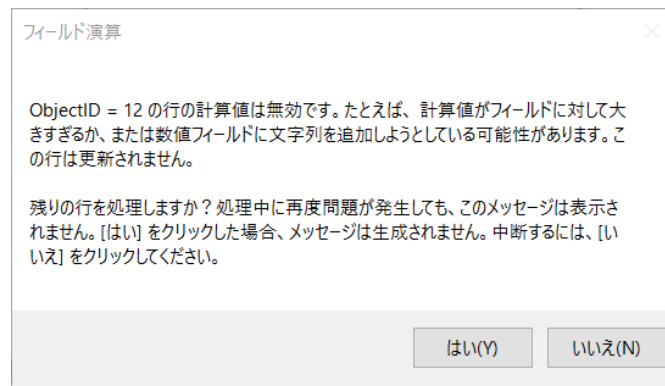
- [属性テーブル] の [テーブル オプション] → [フィールドの追加] → [フィールドの追加]

ダイアログで以下の情報を入力して [OK] をクリックします。

- 名前 : 「MALE」と入力
- タイプ : ドロップダウン リストから [Long Integer] を選択
- 全桁数 : 「10」と入力

- [h27ka15202.MALE] フィールドが作成されていることを確認し、[h27ka15202.MALE] フィールドを右クリックして [フィールド演算] を選択します。
- [編集セッション外で計算を行おうとしています。] というダイアログ が表示されたら [Yes] をクリックします。
- [フィールド演算] ダイアログで、[フィールド] 欄の下部にある「tblT000848C15.txt.T000848002」をダブルクリックしてフィールド名を挿入し、以下のように演算式を記述します。

- [OK] をクリックし、フィールド演算を実行します。
- [ObjectID = 12 の行の計算値は無効です。] というダイアログが出てきますが、[はい] をクリックして演算処理を続けます。



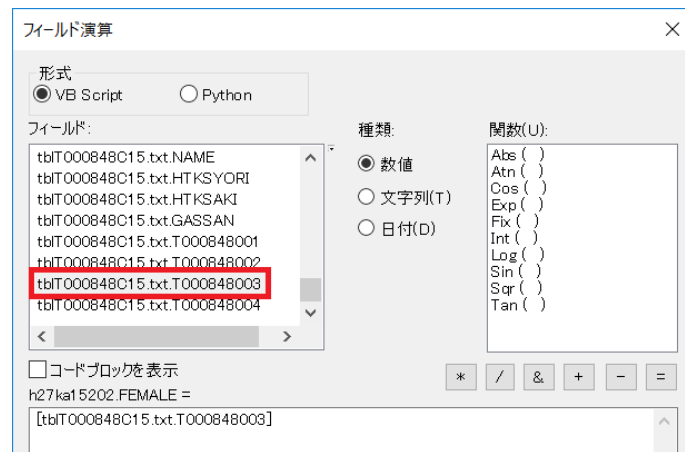
- [処理中に警告が生成されました。] というダイアログが出てくるので、[OK] をクリックします。

上記の 2 つの [ダイアログ] は、「tblIT000848C15.txt.T000848002」フィールドに属性値が入力されていないレコードがあるため演算処理が無効となり、警告として表示されました。

フィールド演算の結果を確認すると、数値型のフィールドになっていることが確認できます (数字が右詰め)。属性値が入力されていなく処理が無効となったレコードは、演算結果として数値の「0」が格納されているので、今後の演習で使用するデータとして問題なく使用することができます。

[h27ka15202.MALE] フィールドと同様の操作を行い、[FEMALE] フィールドを作成します。

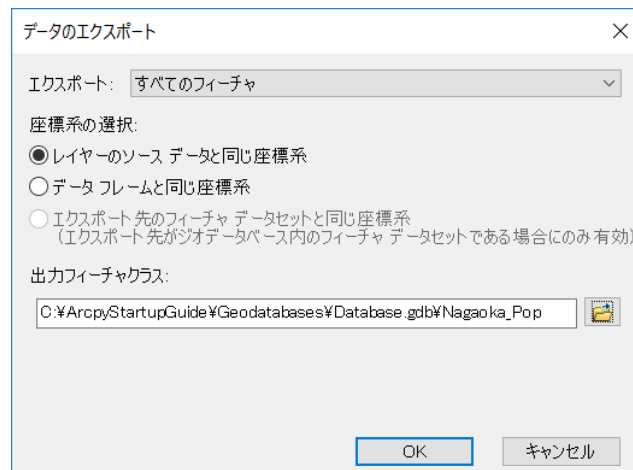
- [属性テーブル] の [テーブル オプション] → [フィールドの追加] → [フィールドの追加] ダイアログで以下の情報を入力して [OK] をクリックします。
 - 名前 : 「FEMALE」と入力
 - タイプ : ドロップダウン リストから [Long Integer] を選択
 - 全桁数 : 「10」と入力
- [h27ka15202.FEMALE] フィールドが作成されていることを確認し、[h27ka15202.FEMALE] フィールドを右クリックして [フィールド演算] を選択します。
- [編集セッション外で計算を行おうとしています。] というダイアログが表示されたら [Yes] をクリックします。
- [フィールド演算] ダイアログで、[フィールド] 欄の下部にある「tblIT000848C15.txt.T000848003」をダブルクリックし、演算式を記述します。



- [OK] をクリックし、フィールド演算を実行します。
- [ObjectID = 12 の行の計算値は無効です。] というダイアログが出てきますが、[はい] をクリックして演算処理を続けます。
- [処理中に警告が生成されました。] というダイアログが出てくるので、[OK] をクリックします。

操作が完了したら、データをエクスポートして新規にレイヤーを作成します。

- [コンテンツ] ウィンドウの「h27ka15202」レイヤー上で右クリックし、[データ] → [データのエクスポート] をクリックし、以下の設定を行い、[OK] をクリックします。
 - エクスポート : [すべてのフィーチャ] を選択
 - 座標系の選択 : [レイヤーのソース データと同じ座標系] をチェック
 - 出力フィーチャクラス :
「C:¥ArcPyStartupGuide¥Geodatabases¥Database.gdb¥Nagaoka_Pop」を入力



- [マップにレイヤーとしてエクスポート データを追加しますか] というダイアログが出てくるので [はい] をクリックします。

これで、長岡市のデータの準備は完了です。次は、警察署のデータをダウンロードします。

ステップ 5: 警察署のデータをダウンロードし、投影法を定義

国土交通省の国土数値情報 ダウンロードサービスから新潟県の警察署のデータをダウンロードします。ダウンロードした「P18-12_15_PoliceStation.shp」には、座標系が定義されていないため、ArcToolbox の投影法の定義ツールを使用して座標系を定義します。ここで作成したデータは、演習 2C で使用します。

- Web ブラウザーで、以下の URL にアクセスします。

国土数値情報 ダウンロードサービス
<http://nlftp.mlit.go.jp/ksj/>

- 「国土数値情報 ダウンロードサービス」サイトで、「3. 地域」列の [警察署] をクリックします。

3. 地域			
<施設>			
国・都道府県の機関	市町村役場等及び公 的集会施設	市区町村役場	公共施設
警察署	消防署	郵便局	医療機関
福祉施設 ■更新	文化施設	学校	都市公園
上水道関連施設	下水道関連施設	廃棄物処理施設	発電施設
燃料給油所	ニュータウン	工業用地	研究機関
地場産業関連施設	物流拠点	集客施設	

- [データのダウンロード (2.各データ詳細)] ページの [ダウンロードするデータの選択] 内から「新潟」をチェックし、[次へ] をクリックします。

ダウンロードするデータの選択							
<input type="checkbox"/> 北海道	<input type="checkbox"/> 青森	<input type="checkbox"/> 岩手	<input type="checkbox"/> 宮城	<input type="checkbox"/> 秋田	<input type="checkbox"/> 山形	<input type="checkbox"/> 福島	<input type="checkbox"/> 茨城
<input type="checkbox"/> 栃木	<input type="checkbox"/> 群馬	<input type="checkbox"/> 埼玉	<input type="checkbox"/> 千葉	<input type="checkbox"/> 東京	<input type="checkbox"/> 神奈川	<input checked="" type="checkbox"/> 新潟	<input type="checkbox"/> 富山
<input type="checkbox"/> 石川	<input type="checkbox"/> 福井	<input type="checkbox"/> 山梨	<input type="checkbox"/> 長野	<input type="checkbox"/> 岐阜	<input type="checkbox"/> 静岡	<input type="checkbox"/> 愛知	<input type="checkbox"/> 三重
<input type="checkbox"/> 滋賀	<input type="checkbox"/> 京都	<input type="checkbox"/> 大阪	<input type="checkbox"/> 兵庫	<input type="checkbox"/> 奈良	<input type="checkbox"/> 和歌山	<input type="checkbox"/> 鳥取	<input type="checkbox"/> 島根
<input type="checkbox"/> 岡山	<input type="checkbox"/> 広島	<input type="checkbox"/> 山口	<input type="checkbox"/> 徳島	<input type="checkbox"/> 香川	<input type="checkbox"/> 愛媛	<input type="checkbox"/> 高知	<input type="checkbox"/> 福岡
<input type="checkbox"/> 佐賀	<input type="checkbox"/> 長崎	<input type="checkbox"/> 熊本	<input type="checkbox"/> 大分	<input type="checkbox"/> 宮崎	<input type="checkbox"/> 鹿児島	<input type="checkbox"/> 沖縄	

- [データのダウンロード (3.ファイルの選択)] ページの「P18-12_15_GML.zip」にチェックを入れ、[次へ] をクリックします。

ファイル名	ファイル容量	年度	測地系	地域
<input checked="" type="checkbox"/> P18-12_15_GML.zip	3.29MB	平成24年	世界測地系	新潟

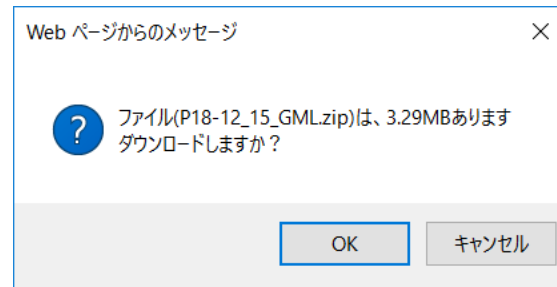
- [アンケートのご協力をお願い] ページに変わるので、回答の記入が完了したら、[回答する] をクリックします。

- [データのダウンロード (4.国土数値情報利用約款)] ページの内容を確認し、[はい] をクリックします。

- [データのダウンロード (5.ダウンロード)] ページの [ダウンロード] をクリックします。

ファイル名	ファイル容量	年度	測地系	地域	ダウンロード
P18-12_15_GML.zip	3.29MB	平成24年	世界測地系	新潟	ダウンロード

- [Webページからのメッセージ] ダイアログが表示されるので [OK] をクリックします。



- 「P18-12_15_GML.zip で行う操作を選んでください」というダイアログが表示されたら、[名前を付けて保存] を選択します。

- 「C:¥ArcPyStartupGuide¥Downloads」フォルダーに「PoliceStation.zip」と名前を付けて保存します。

ダウンロードした「PoliceStation.zip」ファイルを、同じパス内に解凍します。

- 「C:¥ArcPyStartupGuide¥Downloads」フォルダーへ移動し、「PoliceStation.zip」ファイルを右クリックして [すべて展開] をクリックします。

- [圧縮 (ZIP 形式) フォルダーの展開] ダイアログが表示されたら、[完了時に展開されたファイルを表示する] にチェックを入れ、[展開] をクリックします。

- フォルダーが「C:¥ArcPyStartupGuide¥Downloads」フォルダーに解凍されたことを確認します。フォルダーを開くと「P18-12_15_PoliceStation.shp」シェープファイルが格納されていることが確認できます。

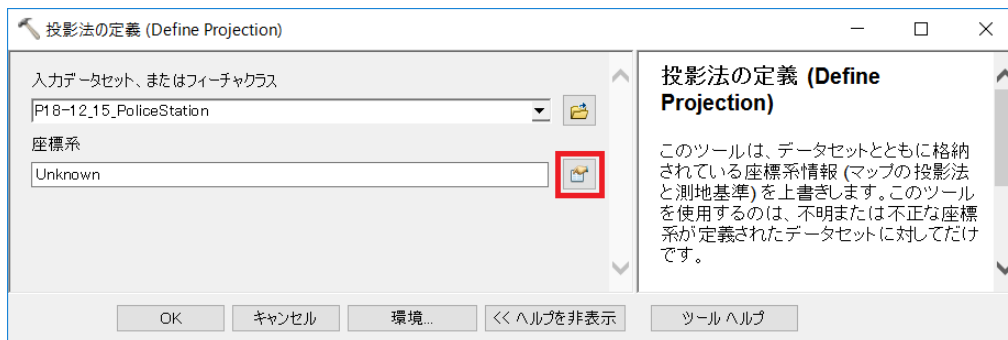
- ArcMap の [カタログ] ウィンドウで、
「C:¥ArcPyStartupGuide¥Downloads¥PoliceStation」フォルダーへ移動し
「P18-12_15_PoliceStation.shp」をドラッグ & ドロップでアクティブ ビューに追加します。

- 追加すると [不明な空間参照] ダイアログが表示されるので、[OK] をクリックします。

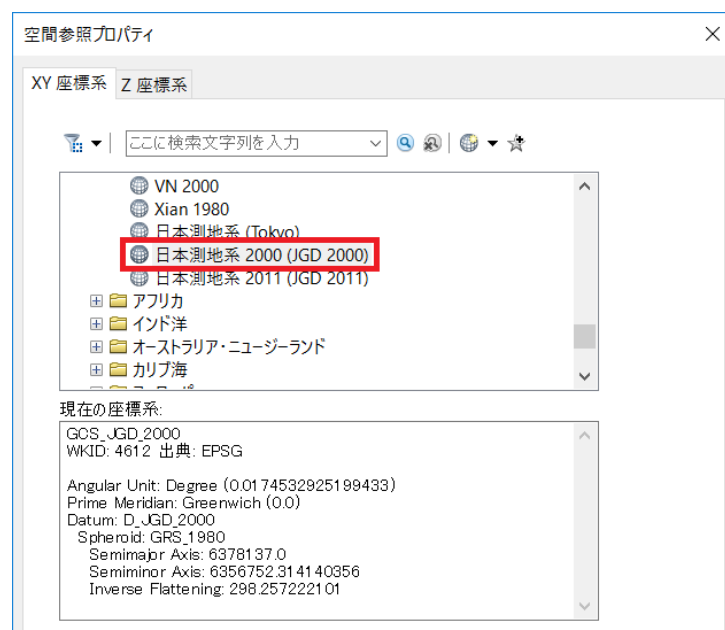
- [ジオプロセッシング] メニューの [ArcToolbox] をクリックします。

- [ArcToolbox] → [データ管理 ツール] → [投影変換と座標変換] → [投影法の定義] をダブルクリックします。

- [入力データセット] にドロップダウン リストから [P18-12_15_PoliceStation] レイヤーを選択します。
- [座標系] で右側のアイコンをクリックします。



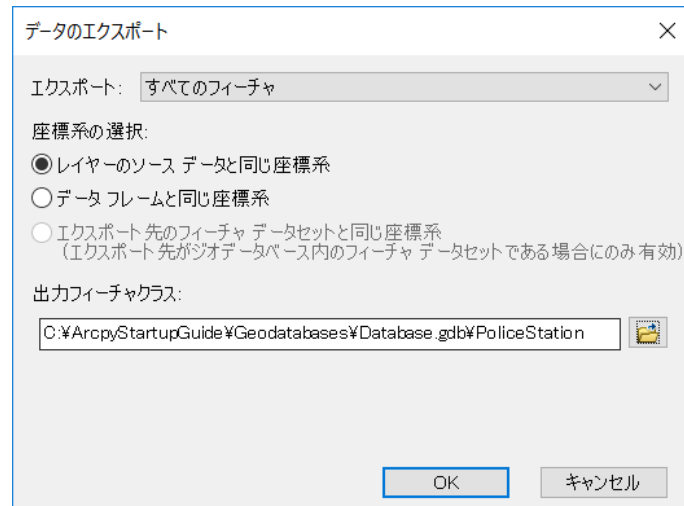
- [空間参照プロパティ] ダイアログで、[地理座標系] → [アジア] → [日本測地系 2000 (JGD 2000)] を選択して [OK] をクリックし、さらに [OK] をクリックしてツールを実行します。



「P18-12_15_PoliceStation」データに、空間参照が定義されました。次に、「P18-12_15_PoliceStation」レイヤーをエクスポートして、フィーチャクラスを作成します。

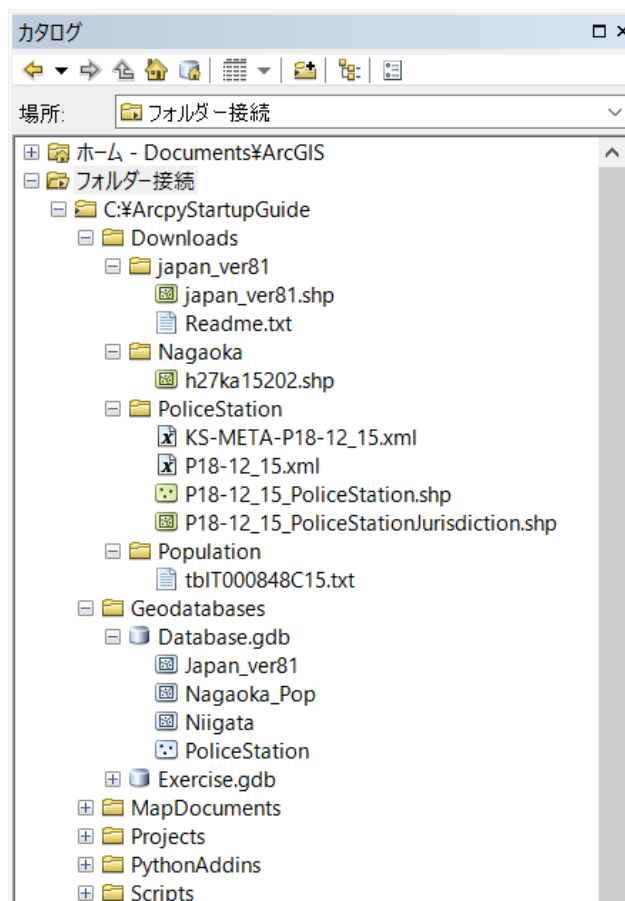
- [コンテンツ] ウィンドウの「P18-12_15_PoliceStation」レイヤー上で右クリックし、[データ] → [データのエクスポート] をクリックし、以下の設定を行い、[OK] をクリックします。

- エクスポート : [すべてのフィーチャ] を選択
- 座標系の選択 : [レイヤーのソース データと同じ座標系] をチェック
- 出力フィーチャクラス :
「C:\%ArcPyStartupGuide%\Geodatabases\Database.gdb\PoliceStation」を入力



- [マップにレイヤーとしてエクスポート データを追加しますか] というダイアログが出でるので [はい] をクリックします。

以上で、演習の事前準備が完了しました。[カタログ] ウィンドウを表示して、下図の状態になっていることを確認してください。



ArcMap を保存せずに、終了します。



第 2 章 ArcPy の基本操作

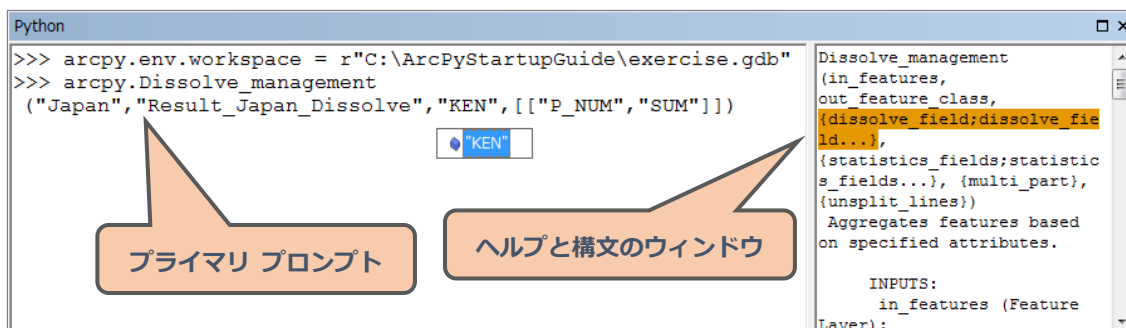
ArcPy の操作方法

この章では、ArcPy もしくは Python 関数を ArcGIS で利用する主な 3 種類の操作方法を学習します。

- ✓ Python ウィンドウ
- ✓ フィールド演算
- ✓ 統合開発環境 (IDE) を使用した ArcPy の実行

Python ウィンドウ

Python ウィンドウは、ArcMap や ArcCatalog 上で対話的に Python の機能を使用するのに、効率的で便利な場所です。



左側のセクションがメインの [Python] ウィンドウ プロンプトで、Python コマンドはここで実行されます。右側のセクションはヘルプと構文のウィンドウで、ここにはツールの実行時に実行メッセージが表示され、コードを入力すると現在のツール、関数、またはクラスについてのヘルプが表示されます。

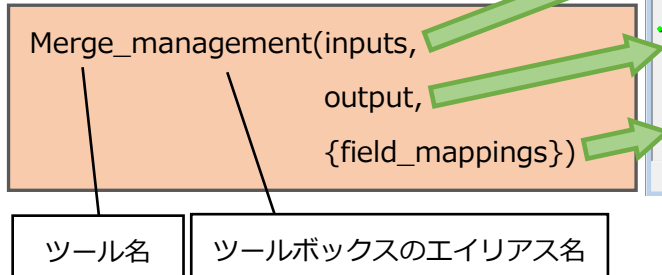
- ✓ Python ウィンドウから関数としてジオプロセッシング ツールを対話的に実行
- ✓ ArcPy サイト パッケージのインポートは不要
- ✓ 既存のスクリプト ファイルを読み込み、実行が可能
- ✓ スクリプト ファイルへの出力、保存が可能

ジオプロセッシング

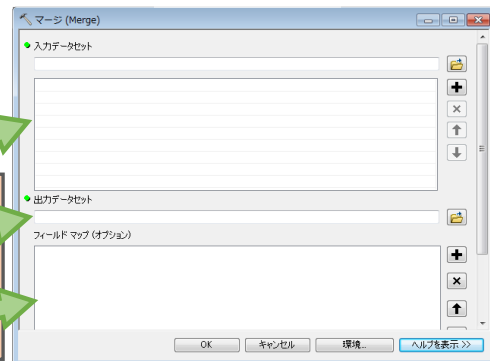
ジオプロセッシングとは、ArcGIS で扱うさまざまなデータを分析したり管理したりすることができるフレームワークです。ジオプロセッシング ツールによってデータの抽出やオーバーレイ、テーブルへの列の追加、最適ルートの計算など、さまざまな処理が実行できます。Python でジオプロセッシング ツールを実行する際は、ツール名とツールボックスのエイリアス名を使用してツールの特定を行います。

<Python スクリプト>

- Buffer_analysis : バッファ ツール
- Project_management : 投影変換 ツール
- Merge_management : マージ ツール



<ツール>



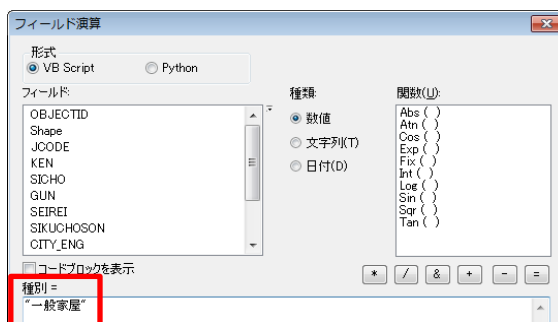
フィールド演算

フィールド演算は、テーブル内の値を使用してすべてのレコードまたは選択したレコードで、単純なものから高度なものまで、さまざまな計算を実行することができる機能です。

単純なフィールド演算式は、[条件式] ボックスに直接入力します。Python スクリプトを使用した繰り返しや条件分岐などの高度な演算式は、[フィールド演算] ダイアログ ボックスの [コード ブロック] ボックスに演算式を記述して実行します。また面積や周長などのジオメトリ演算を、Python を使用することでフィールド演算から計算をすることができます。

単純なフィールド演算

- ✓ テーブル内の値を使用して演算を実行



名称	種別
えすり農林事務所	<NULL>
えすり植物研究所	<NULL>
えすり植物研究所	<NULL>
<NULL>	一般家屋
<NULL>	一般家屋
<NULL>	一般家屋

高度なフィールド演算

- ✓ def 文 (ステートメント) を使用して、独自の関数を作成して演算を実行



条件式で指定するフィールド名は ! (エクスクラメーション マーク) で囲みます。

人口密度	pop_rank
8	Low
83	High
71	High
4	Low
9	Low
79	High
15	Ave
9	Low
6	Low
17	Ave
7	Low
5	Low
78	High
35	Ave

<条件式>

```
poprank(!人口密度!)
```

<コードブロック>

```
def poprank(mitudo):
    if mitudo <= 10:
        return "Low"
    elif mitudo <= 50:
        return "Ave"
    else:
        return "High"
```

統合開発環境（IDE）を使用した ArcPy の実行

統合開発環境（IDE）とは、プログラム開発において用いられるエディター、コンパイラ、リンク、デバッガ、その他の支援ツールなどを統合・統一化した開発環境です。

Python のプログラムはテキスト形式（太字やルビといった文字装飾のない文書）で記述します。プログラムはテキスト エディターで記述することもできますが、IDE を使用するとより容易にプログラムが作成できます。

コードを書くときに使用する「コード」を編集する専用の機能を持った、プログラミング専用のテキスト エディターというのがあり、そのアプリケーションを使用することで容易にプログラムを書くことができます。

IDE を使用して ArcPy を実行する利点

- ✓ ArcMap を起動しなくても GIS データを編集することができる
- ✓ IDE に搭載されているデバッガを使用することで、スクリプトの実行前に問題を検出することができる
- ✓ 記述したスクリプトを保存することができ、同様の処理を繰り返し実行することができる

IDE を使用して スクリプト ファイルを作成する際の注意点

- ✓ マルチバイト文字の記述を行うために文字コードの指定をする必要がある
 - スクリプトの 2 行目までに、入力するテキストの文字コードに応じてコメント文で指定

`#coding:cp932` または `#coding:Shift_JIS` または `#coding:utf-8`

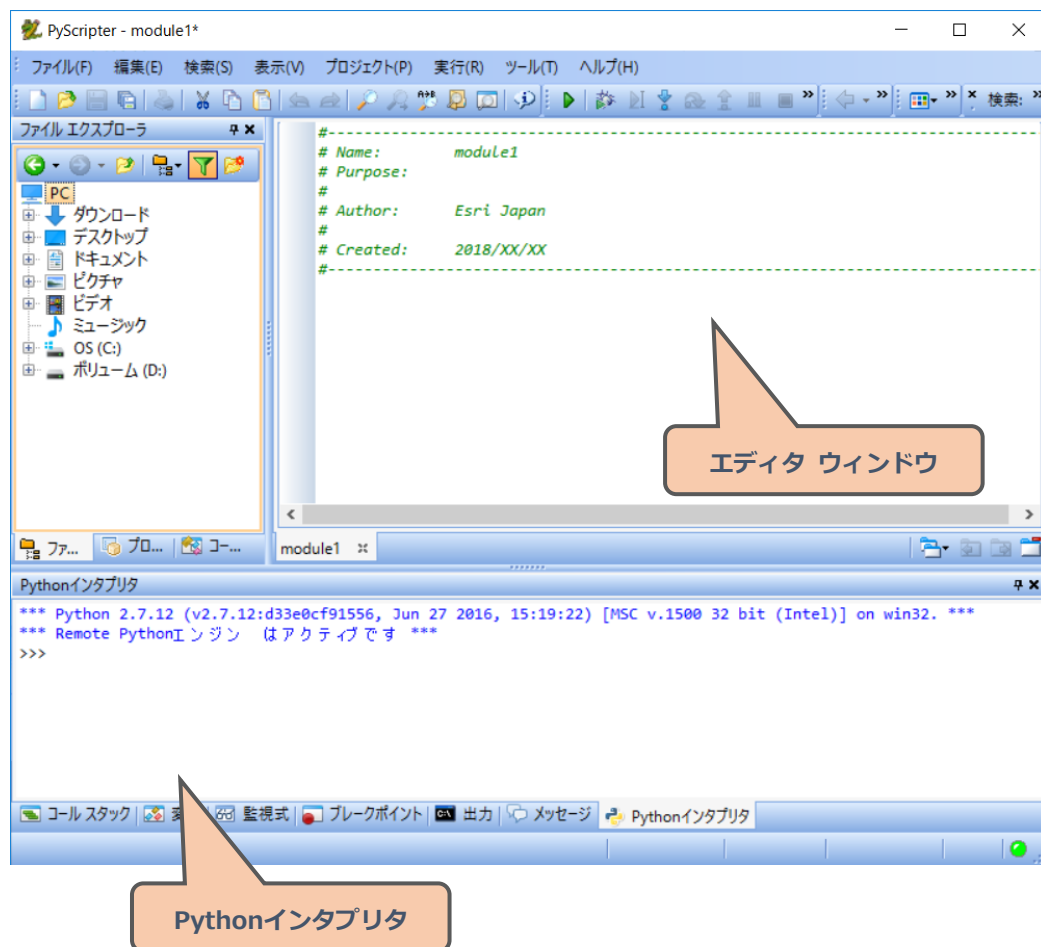
- 指定する文字コードはファイルに依存
 - 指定する文字コードとスクリプト ファイルの文字コードの一致が必要
 - 大文字と小文字、ハイフンとアンダーバーの違いは Python が考慮
- ✓ ArcPy サイト パッケージのインポートが必要

PyScripter

PyScripter には以下の特徴があります。

- ✓ 日本語のユーザー インタフェース (UI) で表示できる
- ✓ オート インデント機能、文字の自動色分け、入力補完、デバッグ機能、構文チェック機能などの Python 用の開発環境が整っている

PyScripter は下図のように、複数のウィンドウで構成されています。



演習 2A: [Python] ウィンドウによる操作

この演習では、[Python] ウィンドウから [ディゾルブ] ツールを実行します。

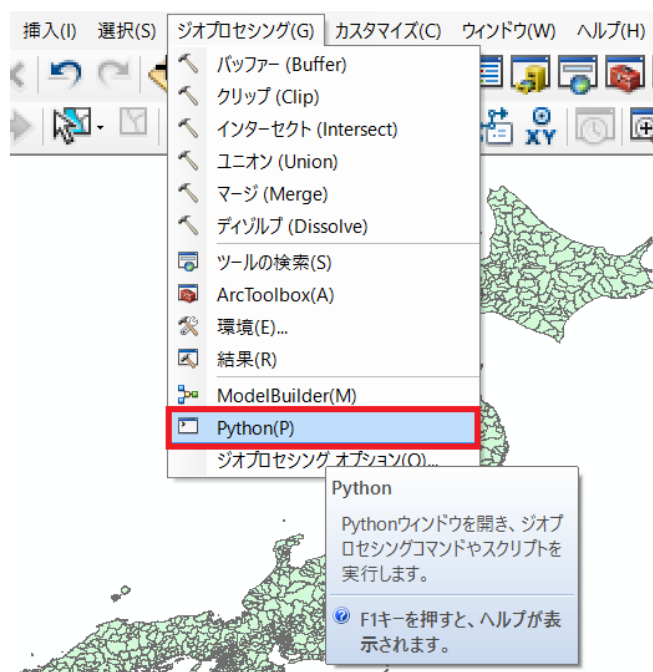


演習時間 : 10 分

ステップ 1: [ディゾルブ] ツールの実行

このステップでは、市区町村界レイヤー (Japan_ver81) 内のポリゴン フィーチャを都道府県ごとに集約し、新規にフィーチャ レイヤーを作成します。また [P_NUM](人口数) フィールドの合計値を都道府県ごとに取得します。

- ArcMap を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcMap 10.7])。
- [ArcMap - はじめに] ダイアログが表示されたら [キャンセル] をクリックします。
- [ツール] ツールバーの [データの追加] ボタンをクリックし、[データの追加] ダイアログを使用して以下のデータをマップに追加します。
 - C:\¥ArcPyStartupGuide¥Geodatabases¥Database.gdb¥Japan_ver81
- [ジオプロセッシング] メニュー → [Python] をクリックし、[Python] ウィンドウを開きます。



[Python] ウィンドウが起動したら、[ディゾルブ] ツールを実行するためのスクリプトコードを記述します。

ディゾルブとは、指定した属性に基づいてフィーチャを集約する機能です。集約されるフィーチャの属性は、さまざまな統計情報（指定されたフィールドの最大値を検出する [MAX] や、平均を計算する [MEAN] など）を使用して、集計したり説明したりできます。

まず、ジオプロセシング ツールの実行に使用するワークスペースを指定するためのコードを入力します。

- [ツール] ツールバーの [カタログ] ボタンをクリックし、[カタログ] ウィンドウを開きます。
- [カタログ] ウィンドウで「C:\ArcpyStartupGuide\Geodatabases\Database.gdb」へ移動します。
- [Python] ウィンドウ プロンプトで、「arcpy.env.workspace」を記述し、[カタログ] ウィンドウから「Database.gdb」を [Python] ウィンドウにドラッグ & ドロップで追加し、以下の通りにコードを記述します。

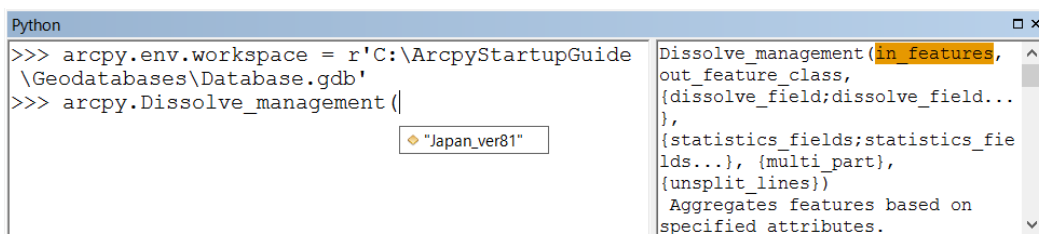
```
>>> arcpy.env.workspace = r"C:\ArcPyStartupGuide\Geodatabases\Database.gdb"
```



ワークスペースには、実行するデータが保存されているフォルダーやジオデータベースの「パス」を選択します。ワークスペースを指定することで、入力するデータのパスを短縮することができます。

- 次のコードを記述するために、Enter キーを押します。
- [ディゾルブ] ツールを実行するための、以下のコードを 1 行で記述します。

```
>>> arcpy.Dissolve_management(
    ("Japan_ver81", "Japan_Dissolve", "KEN", [{"P_NUM",
    "SUM"}])
```



[Python] ウィンドウでは、関数の括弧を入力すると、[Python] ウィンドウの右側にあるヘルプパネルに、入力する必要のある関数の引数や説明について表示されます。また、入力補助機能として、マップに追加されているレイヤーの一覧から、ツールの引数を選択することができます。

```
Python
>>> arcpy.env.workspace = r"C:\ArcPyStartupGuide\Geodatabases\Database.gdb"
>>> arcpy.Dissolve_management(["Japan_ver81",
"Japan_Dissolve", "KEN", [{"P_NUM", "SUM"}]])
```

□ Enter キーを押し、関数を実行します。

しばらくすると、処理が完了し、レイヤーがマップに追加されます。



□ 新規に作成されたレイヤー (Japan_Dissolve) が、データ フレームに追加されたことを確認します。

□ [コンテンツ] ウィンドウの「Japan_Dissolve」レイヤーを右クリックし、[属性テーブルを開く] をクリックします。

テーブル

Japan_Dissolve

	OBJECTID *	Shape *	KEN	SUM_P_NUM	Shape_Length	Shape_Area
▶	1	Polygon	愛知県	7509636	8.27996	0.510944
	2	Polygon	愛媛県	1415997	17.397462	0.551917
	3	Polygon	茨城県	2970231	6.191827	0.611795
	4	Polygon	岡山県	1933781	8.863161	0.701777
	5	Polygon	沖縄県	1461231	16.313503	0.205177
	6	Polygon	岩手県	1289470	11.636805	1.601766
	7	Polygon	岐阜県	2076195	7.76215	1.058851
	8	Polygon	宮崎県	1128078	8.067745	0.739755
	9	Polygon	宮城県	2324466	12.450016	0.751432
	10	Polygon	京都府	2574842	7.565276	0.456905
	11	Polygon	熊本県	1810343	13.806394	0.711961
	12	Polygon	群馬県	2005320	5.803486	0.639995

1 (0 / 47 選択)

Japan_Dissolve

都道府県ごとにフィーチャが集約され、[P_NUM] フィールドの合計値を取得していることが確認できます。この演習で行った操作は、ArcToolbox の [ディゾルブ] ツールを実行するのと同等の関数を呼び出しました。

- ArcMap の [ファイル] メニューから [名前を付けて保存] を選択し、
「C:¥ArcPyStartupGuide¥MapDocuments」フォルダーに「Japan_Dissolve.mxd」と名前を付けて保存してください。
- ArcMap を終了します。

演習 2B: フィールド演算による操作

この演習では、Python を使用した単純なフィールド演算と高度なフィールド演算の 2 種類のフィールド演算を実行します。

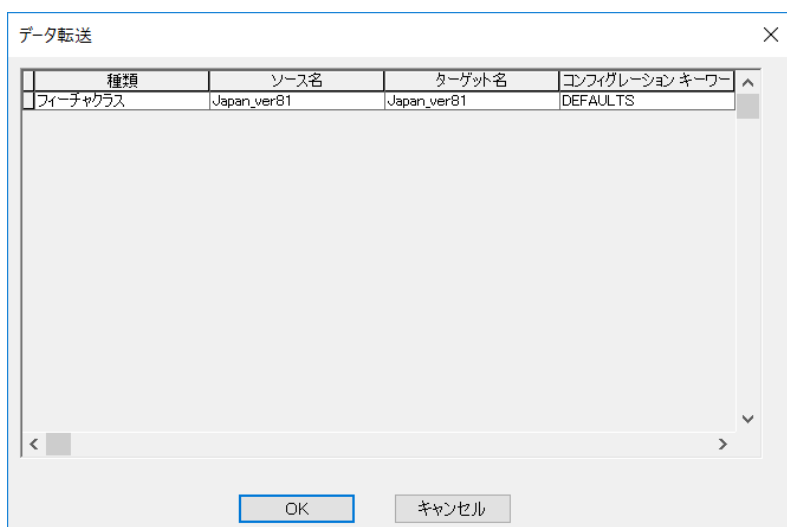


演習時間 : 10 分

ステップ 1: 単純なフィールド演算の実行

このステップでは、「Japan_ver81」レイヤーに新規にフィールドを追加して、面積の計算をフィールド演算から実行します。また、人口数の統計データを持つ [P_NUM] フィールドを使用して各市区町村の人口密度を算出し、階級区分図を作成します。

- ArcMap を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcMap 10.7])。
- [ArcMap - はじめに] ダイアログが表示されたら [キャンセル] をクリックします。
- [ツール] ツールバーの [カタログ] ボタンをクリックし、[カタログ] ウィンドウを開きます。
- 「C:\¥ArcPyStartupGuide¥Geodatabases¥Database.gdb」へ移動し、[Japan_ver81] フィーチャクラス上で右クリックし、[コピー] をクリックします。
- [カタログ] ウィンドウの「Exercise.gdb」上で右クリックし [貼り付け] をクリックします。
- [データ転送] ダイアログが表示されるので、[OK] をクリックします。



- 「Exercise.gdb」に「Japan_ver81」フィーチャクラスが格納されたことを確認します。

- 「Exercise.gdb」内の「Japan_ver81」フィーチャクラス上で右クリックし、[名前の変更] を選択します。
- 名前を「Japan_PopDensity」とします。
- 「Exercise.gdb」内の「Japan_PopDensity」フィーチャクラスをドラッグ & ドロップで アクティブ ビューに追加します。
- [コンテンツ] ウィンドウの「Japan_PopDensity」レイヤーを右クリックし、[属性テーブルを開く] を選択します。
- 属性テーブルの [テーブル オプション] → [フィールドの追加] を選択し、[フィールドの追加] ダイアログで以下の情報を入力して [OK] をクリックします。
 - 名前 : 「AREA」と入力
 - タイプ : ドロップダウン リストから [Double] を選択

フィールド プロパティ	
エイリアス	
NULL 値を許可	はい
デフォルト値	

- [AREA] フィールドが作成されていることを確認し、[AREA] フィールドを右クリックして [フィールド演算] を選択します。
- [編集セッション外で計算を行おうとしています。] というダイアログ が表示されたら [Yes] をクリックします。
- [フィールド演算] ダイアログが表示されたら、上部にある [形式] を [Python] に変更します。

- [フィールド] 内のリストから、[Shape] フィールドをダブルクリックし、入力された文字列の続きに、以下の通りに演算式を記述します。

!Shape.area@SQUAREKILOMETERS!

Shape_Area
AREA

☐ コードブロックを表示

AREA =

!Shape.area@SQUAREKILOMETERS!

math.acos()
math.acosh()
math.asin()

* / & + - =

@SQUAREKILOMETERS は、単位を平方キロメートルに変換するためのトークンです。

- [OK] をクリックし、フィールド演算を実行します。
- [AREA] フィールドに面積が計算されていることが確認できたら、次に人口密度を算出します。
- 属性テーブルの [テーブル オプション] → [フィールドの追加] をクリックし、[フィールドの追加] ダイアログで以下の情報を入力して [OK] をクリックします。
 - 名前 : 「POPDENSITY」と入力
 - タイプ : ドロップダウン リストから [Double] を選択
 - [POPDENSITY] フィールドが作成されていることを確認し、[POPDENSITY] フィールドを右クリックして [フィールド演算] を選択します。
 - [編集セッション外で計算を行おうとしています。] というダイアログが表示されたら [Yes] をクリックします。
 - [フィールド演算] ダイアログが表示されたら、[形式] が [Python] に選択されていることを確認します。
 - [フィールド] 内のリストから、[P_NUM] フィールドと [AREA] フィールドをダブルクリックし、以下の通りに演算式を記述します。

!P_NUM!/!AREA!

AREA
POPDENSITY

☐ コードブロックを表示

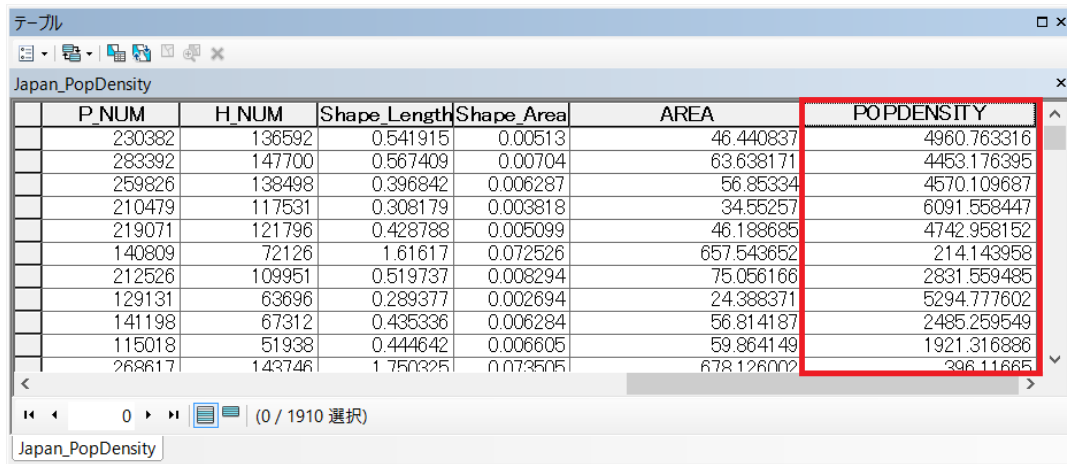
POPDENSITY =

!P_NUM!/!AREA!

math.acos()
math.acosh()
math.asin()

* / & + - =

- [OK] をクリックし、フィールド演算を実行します。



	P_NUM	H_NUM	Shape_Length	Shape_Area	AREA	POPDENSITY
	230382	136592	0.541915	0.00513	46.440837	4960.763316
	283392	147700	0.567409	0.00704	63.638171	4453.176395
	259826	138498	0.396842	0.006287	56.85334	4570.109687
	210479	117531	0.308179	0.003818	34.55257	6091.558447
	219071	121796	0.428788	0.005099	46.188685	4742.958152
	140809	72126	1.61617	0.072526	657.543652	214.143958
	212526	109951	0.519737	0.008294	75.056166	2831.559485
	129131	63696	0.289377	0.002694	24.388371	5294.777602
	141198	67312	0.435336	0.006284	56.814187	2485.259549
	115018	51938	0.444642	0.006605	59.864149	1921.316886
	268817	143746	1.750325	0.073505	678.126002	396.11665

[POPDENSITY] フィールドに、人口密度（人/km²）が計算されました。次に、[POPDENSITY] フィールドを使用して階級区分図を作成します。

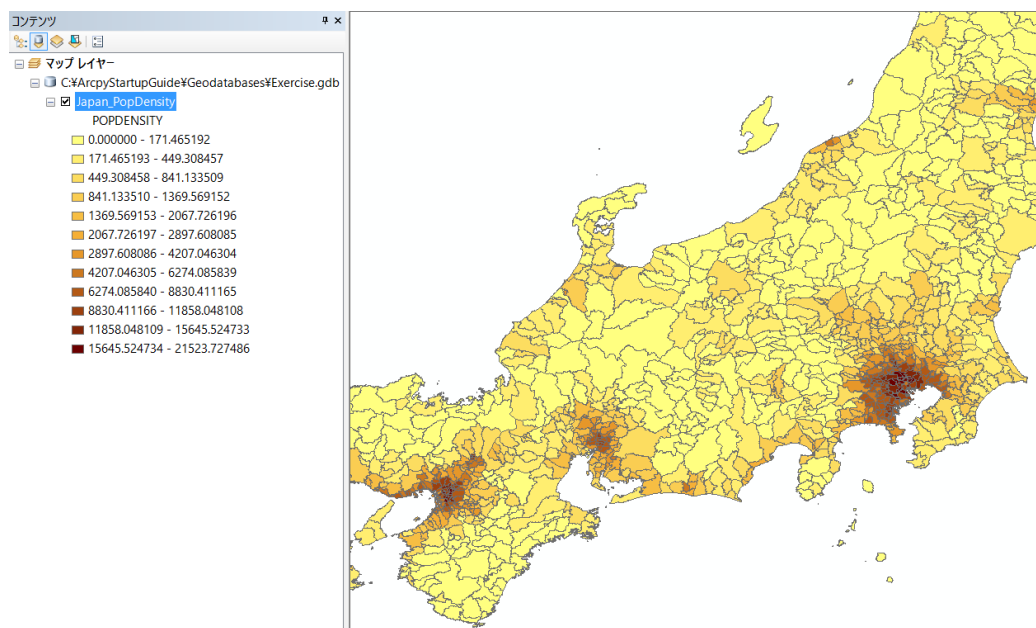
- 開いている属性テーブルを閉じます。
- [コンテンツ] ウィンドウの「Japan_PopDensity」レイヤーを右クリックし、プロパティを開きます。
- [レイヤー プロパティ] ダイアログが表示されたら、[シンボル] タブをクリックします。
- [表示] で [数値分類] をクリックし、[等級色] を選択します。

デフォルトで、[等級色] が選択された状態になっています。

- [値] のドロップダウン リストで、「POPDENSITY」フィールドを選択します。

選択したフィールドの属性値が 5 つに分類されます。[カラー ランプ] にはグラデーションカラーが設定され、各クラスに適切な色が割り振られます。クラスの範囲は均等ではなく、属性値の特徴（値同士の隔たりや偏りの度合い）を認識するアルゴリズムによって決定されます。

- [分類] に [自然分類] が選択されていることを確認し、[クラス] のドロップダウン リストから [12] を選択します。
- [カラー ランプ] の色の上で右クリックし、[グラフィック ビュー] のチェックを外します。[カラー ランプ] が色での表示から名前での表示に変わります。
- [カラー ランプ] を「黄から濃い赤」に設定します。
- [レイヤー プロパティ] ダイアログで、[適用] をクリックし、[OK] をクリックします。



各市区町村の人口密度（人/km²）の値を基に、フィーチャが濃淡で表示されます。

- ArcMap の [ファイル] メニューから [名前を付けて保存] を選択し、
「C:\ArcPyStartupGuide\MapDocuments」フォルダーに
「Japan_PopDensity_GraduatedColors.mxd」と名前を付けて保存してください。

ステップ 2: 高度なフィールド演算の実行

このステップでは、長岡市の各町丁字において男女どちらの人口が多いかを、Python の条件分岐構文「if...elif...else」を使用してフィールド演算を実行し、新規に追加した「RECLASS」フィールドに再分類した値を出力します。

- ArcMap → [ツール] ツールバー → [新規] ボタンをクリックし、[新規ドキュメント] ダイアログで [OK] をクリックし、空のマップ ドキュメントを開きます。
- [カタログ] ウィンドウから「C:\%ArcPyStartupGuide%Geodatabases%Database.gdb」へ移動し、「Nagaoka_Pop」フィーチャクラス上で右クリックし、[コピー] をクリックします。
- [カタログ] ウィンドウの「Exercise.gdb」上で右クリックし [貼り付け] をクリックします。
- [データ転送] ダイアログが表示されるので、[OK] をクリックします。
- 「Exercise.gdb」に「Nagaoka_Pop」フィーチャクラスが格納されたことを確認します。
- 「Exercise.gdb」内の「Nagaoka_Pop」フィーチャクラス上で右クリックし、[名前の変更] を選択します。
- 名前を「Nagaoka_Reclass」とします。
- 「Exercise.gdb」内の「Nagaoka_Reclass」フィーチャクラスをドラッグ & ドロップでアクティブ ビューに追加します。
- [コンテンツ] ウィンドウの「Nagaoka_Reclass」レイヤーを右クリックし、[属性テーブルを開く] を選択します。
- [属性テーブル] の [テーブル オプション] → [フィールドの追加] を選択し、[フィールドの追加] ダイアログで以下の情報を入力して [OK] をクリックします。
 - 名前 : 「RECLASS」と入力
 - タイプ : ドロップダウン リストから [Text] を選択
 - 長さ : 「10」とし入力

フィールド プロパティ	
エイリアス	
NULL 値を許可	はい
デフォルト値	
長さ	10

- [RECLASS] フィールドが作成されていることを確認し、[RECLASS] フィールドを右クリックして [フィールド演算] を選択します。
- [編集セッション外で計算を行おうとしています。] というダイアログが表示されたら [Yes] をクリックします。
- [フィールド演算] ダイアログが表示されたら、上部にある [形式] を [Python] に変更します。
- [コードブロックを表示] にチェックを入れます。
- [Pre-Logic Script Code:] ボックスに以下のコードを記述します。

```
def reclass(male, female):
    ex = male-female
    if ex > 0:
        return "male"
    elif ex < 0:
        return "female"
    else:
        return "even"
```

「def」は、関数を定義するための文（ステートメント）です。「def」文の後に関数名を定義し、引数を括弧で囲んで定義します。



Python では、インデントも言語を構成する一部なので同じスペース数でインデントを挿入してください。

- [RECLASS =] のボックスに以下のコードを記述します。

```
reclass(!MALE!, !FEMALE!)
```

- 以下の通りに演算式の記述が完了したら、[OK] をクリックしフィールド演算を実行します。

☒ コードブロックを表示

Pre-Logic Script Code:

```
def reclass(male, female):
    ex = male - female
    if ex > 0:
        return "male"
    elif ex < 0:
        return "female"
```

RECLASS =

```
reclass(!MALE, !FEMALE)
```

[フィールド演算について](#)

消去(C) 読み込み(L)... 保存(S)...

OK キャンセル



エラーメッセージが表示された場合は、Python の構文が正しく記述されているかをもう一度確認し、フィールド演算を実行してください。特に、インデントがそろっていることを確認してください。

[RECLASS] フィールドに、フィールド演算によって新しく算出された値が追加されていることが確認できました。

テーブル

Nagaoka_Reclass

	T000848002	T000848003	T000848004	Shape_Length	Shape_Area	RECLASS
	111	98	109	823.369811	37821.304195	male
	312	368	317	1134.539147	64566.903621	female
	76	88	110	2584.580407	216792.395081	female
	151	144	141	881.614223	47010.364354	male
	104	107	87	945.156036	42314.772065	female
	131	141	111	1135.214187	50044.461292	female
	195	211	156	1128.403496	61966.752831	female
	164	213	155	918.124673	48024.74061	female
	178	186	164	1244.067543	64447.244062	female
	-	-	-	1662.439947	120297.16914	even
	-	-	-	-	-	-

0

(0 / 956 選択)

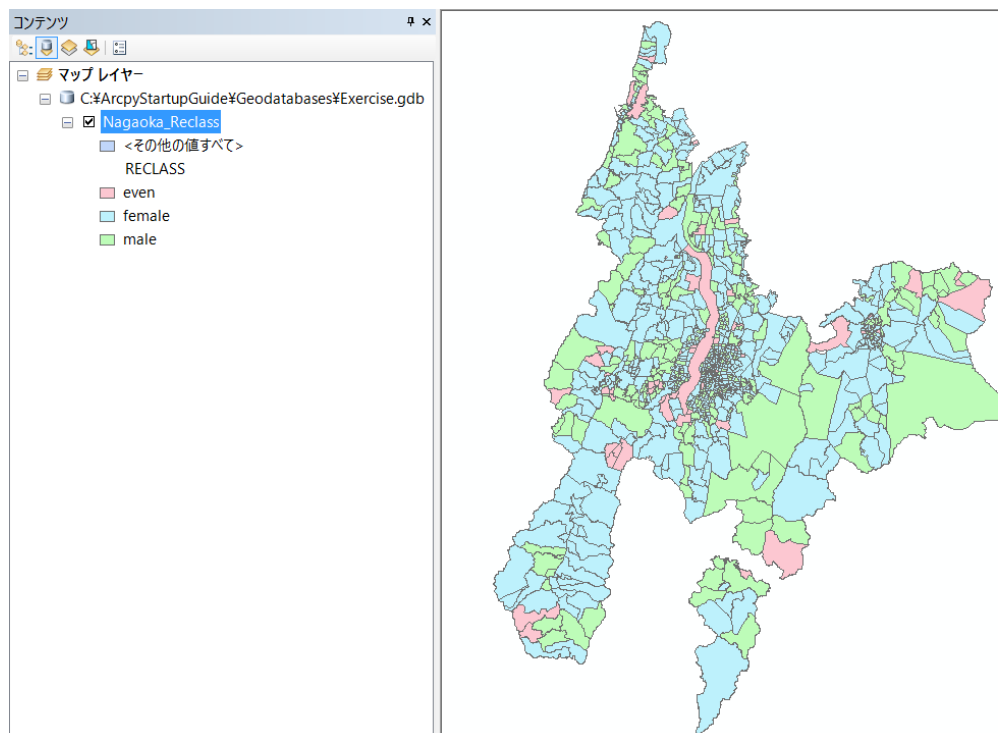
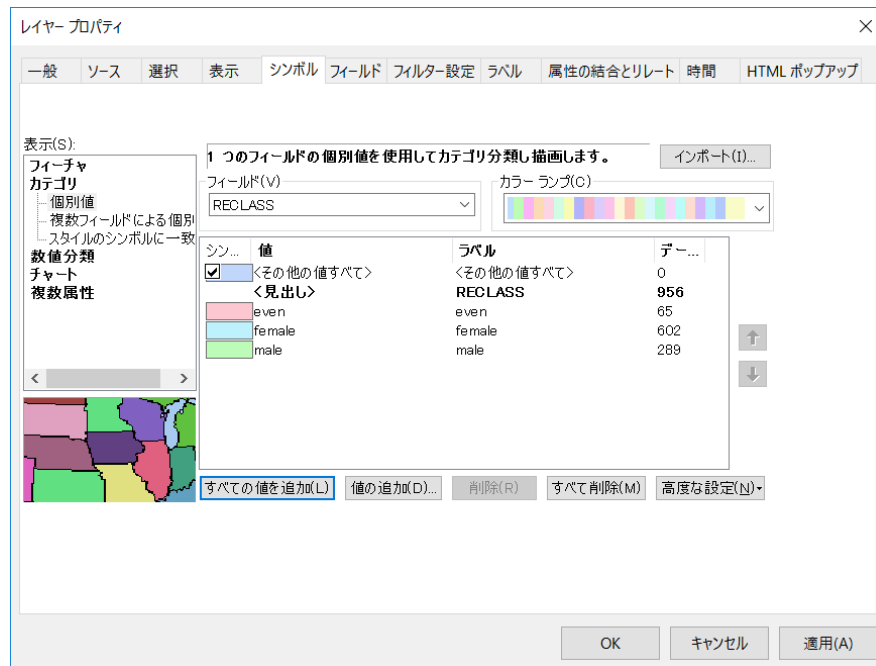
Nagaoka_Reclass

[MALE] フィールドと [FEMALE] フィールドを使用して算出した値が、[RECLASS] フィールドに格納され、「male」「female」「even」の 3 クラスに分類されていることが確認できます。

次に、「Nagaoka_Reclass」レイヤーの [RECLASS] フィールドを個別値で分類します。

- 開いている属性テーブルを閉じます。
- [コンテンツ] ウィンドウの「Nagaoka_Reclass」レイヤーを右クリックし、[プロパティ] を選択します。
- [レイヤー プロパティ] ダイアログが表示されたら、[シンボル] タブをクリックします。
- [表示] で [カテゴリ] をクリックし、[個別値] を選択します。
- [フィールド] で「RECLASS」を選択します。

- [すべての値を追加] をクリックします。
- [レイヤー プロパティ] ダイアログで、[適用] をクリックし、[OK] をクリックします。



- ArcMap の [ファイル] メニューから [名前を付けて保存] を選択し、
「C:\ArcPyStartupGuide\MapDocuments」 フォルダーに
「Nagaoka_Reclass_UniqueValues.mxd」と名前を付け、[保存] をクリックします。

演習 2C: 統合開発環境 (IDE) による操作

この演習では、PyScripter を使用して [バッファー] ツールと [クリップ] ツールを実行します。

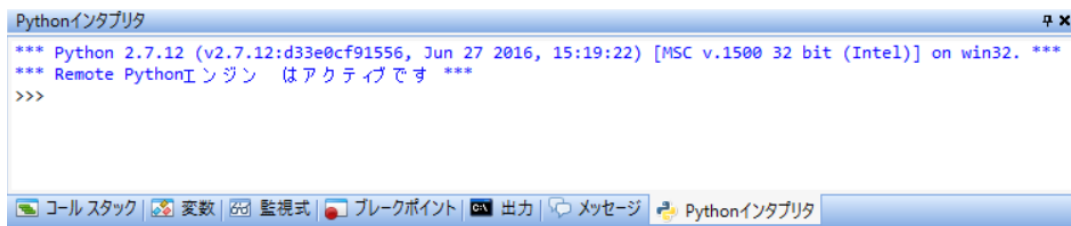


演習時間 : 10 分

ステップ 1: Python インタプリタによる操作

このステップでは、PyScripter の [Pythonインタプリタ] ウィンドウを使用し、警察署のポイント フィーチャクラス (PoliceStation) から 1500 m バッファーを作成します。Python インタプリタを使用すると、コードを逐次実行できます。

- PyScripter を起動します (デスクトップのショートカットアイコン、または Windows の [スタート] プログラム → [PyScripter] → [PyScripter for Python 2.7])。
- [Pythonインタプリタ] ウィンドウ上に、>>> プロンプトが表示されていることを確認します。表示されていない場合は、Enter キーを押します。



- 以下のコードを記述し、Enter キーを押します。

```
>>> import arcpy
```

ArcPy サイト パッケージがインポートされました。

- 「C:\ArcPyStartupGuide\Geodatabases\Database.gdb」をワークスペースに設定する以下のコードを記述し、Enter キーを押します。

```
>>> arcpy.env.workspace =  
r"C:\ArcPyStartupGuide\Geodatabases\Database.gdb"
```

次に、[バッファー] ツールへ入力するパラメーターの変数の設定をします。

- 入力フィーチャクラスを設定する変数を設定します。以下のコードを記述し、Enter キーを押します。

```
>>> inputFC = "PoliceStation"
```

- 出力フィーチャクラスを設定する変数を設定します。以下のコードを記述し、Enter キーを押します。

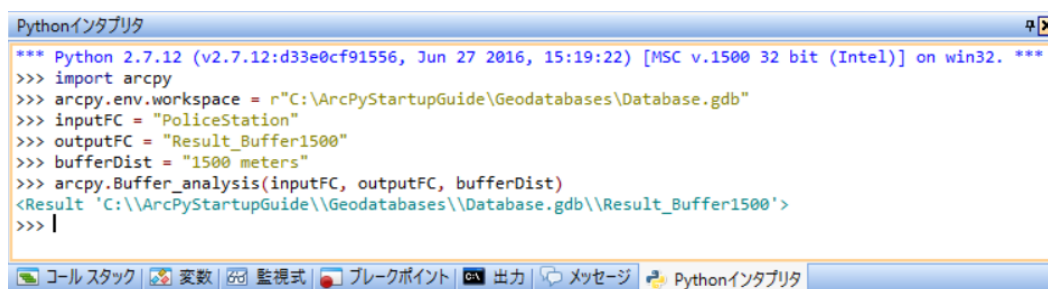
```
>>> outputFC = "Result_Buffer1500"
```

- バッファ処理される入力フィーチャの周囲の距離を設定する変数を設定します。以下のコードを記述し、Enter キーを押します。

```
>>> bufferDist = "1500 meters"
```

- [バッファ] ツールを実行します。以下のコードを記述し、Enter キーを押します。

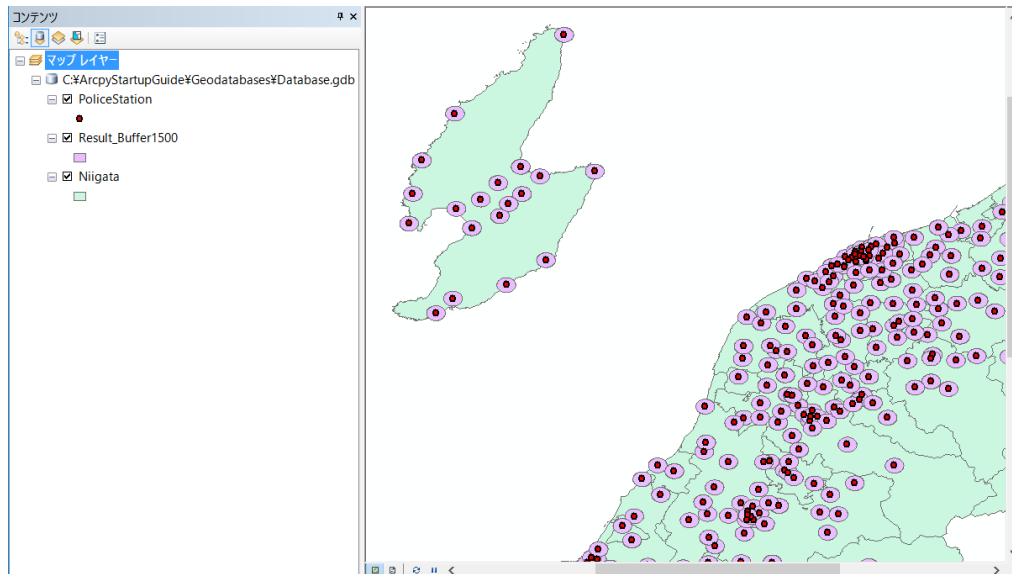
```
>>> arcpy.Buffer_analysis(inputFC, outputFC, bufferDist)
```



現在は「C:\ArcPyStartupGuide\Geodatabases\Database.gdb」ファイル ジオデータベースをワークスペースに指定しています。そのため、Buffer_analysis 関数の第一引数には「C:\ArcPyStartupGuide\Geodatabases\Database.gdb\PoliceStation」が設定され、第二引数には「C:\ArcPyStartupGuide\Geodatabases\Database.gdb\Result_Buffer1500」と記述されていることと等しくなります。

処理が完了したら、ArcMap を起動して、新規に「Result_Buffer1500」ポリゴン フィーチャクラスが作成されていることを確認します。

- ArcMap を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcMap 10.7])。
- [ArcMap - はじめに] ダイアログが表示されたら [キャンセル] をクリックします。
- [ツール] ツールバーの [データの追加] ボタンをクリックし、[データの追加] ダイアログを使用して以下の 3 つのデータをマップに追加します。
 - C:\ArcPyStartupGuide\Geodatabases\Database.gdb\PoliceStation
 - C:\ArcPyStartupGuide\Geodatabases\Database.gdb\Result_Buffer1500
 - C:\ArcPyStartupGuide\Geodatabases\Database.gdb\Niigata



- [ファイル] メニューから [名前を付けて保存] を選択し
「C:\ArcPyStartupGuide\MapDocuments」フォルダーに、
「PoliceStation_Buffer1500.mxd」と名前を付けて保存してください。
- ArcMap を終了します。

ステップ 2: Python モジュールによるプログラムの実行

このステップでは、PyScripter の Python モジュールを使用してステップ 1 で作成した警察署ポイントのバッファー (Result_Buffer1500) を、新潟県のポリゴン フィーチャクラス (Niigata) でクリップし、新規に出力する操作を行います。Python モジュールを使用すると、一連のコードをコンパイルして実行できます。

- PyScripter が開いていなければ、起動します (デスクトップのショートカットアイコン、または Windows の [スタート] プログラム → [PyScripter] → [PyScripter for Python 2.7])。
- PyScripter の Python モジュールに以下のコードを記入します (コメント部分 (#) は記入せず、太字のコード部分のみ記入してもスクリプトは実行できます)。

```
# coding:cp932
#
#-----
# Name:      Clip_Buffer1500.py
# Purpose:   作成した1500 m バッファーを Niigata ポリゴン フィーチャクラスでクリップ
#            します。
#
# Author:
#
# Created:   <今日の日付>
#-----

# ArcPy サイト パッケージのインポート
import arcpy
```

```
# ワークスペースの設定
arcpy.env.workspace = r"C:\¥ArcPyStartupGuide¥Geodatabases¥Database.gdb"

# クリップツールへ入力するパラメーターの変数の設定
# クリップ対象のフィーチャクラス
inputFC = "Result_Buffer1500"

# 入力フィーチャのクリップに使用するフィーチャクラス
clipFC = "Niigata"

# 出力フィーチャクラス
outputFC = "Result_Clip_Buffer1500"

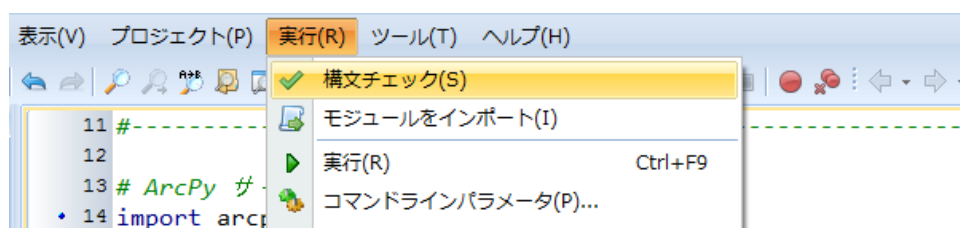
# 【解析】 ツール ボックスの [クリップ] ツールを実行
arcpy.Clip_analysis(inputFC, clipFC, outputFC)

print "実行が完了しました。"
```



最初の行に「# coding:cp932」を忘れずに入力してください。

- PyScripter の [実行] メニューから [構文チェック] をクリックし、入力したコードを確認します。



PyScripter の下部ウィンドウが [メッセージ] ウィンドウに切り替わり、コードに問題がなければ「module1 のシンタックスは OK です!」というメッセージが表示されます。記述した Python コードに構文エラーが存在しないことが分かりました。

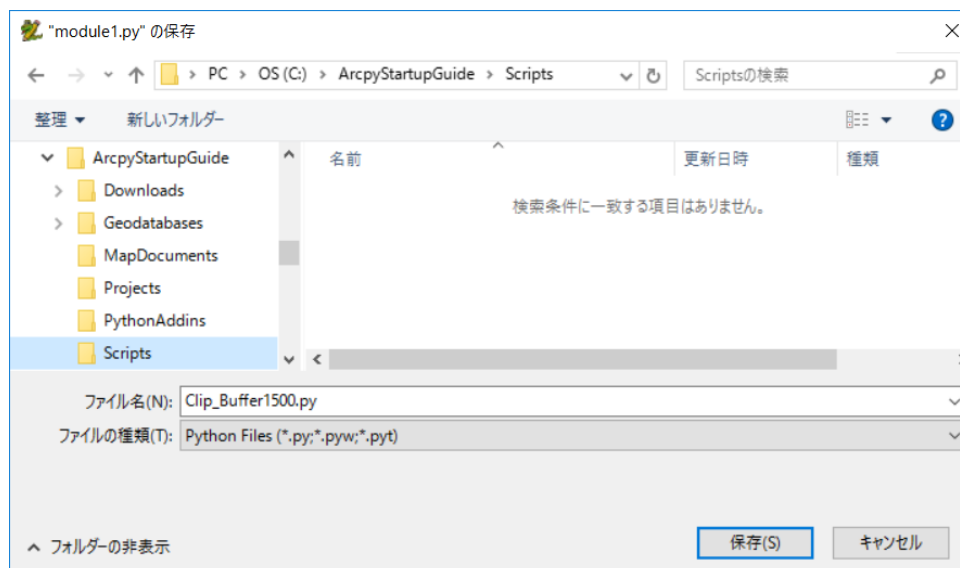
- [デバック] ツールバー → [実行] ボタンをクリックし、スクリプトを実行します。



しばらくすると、コードの実行が完了し [Pythonインタプリタ] ウィンドウに「実行が完了しました。」メッセージが表示されます。

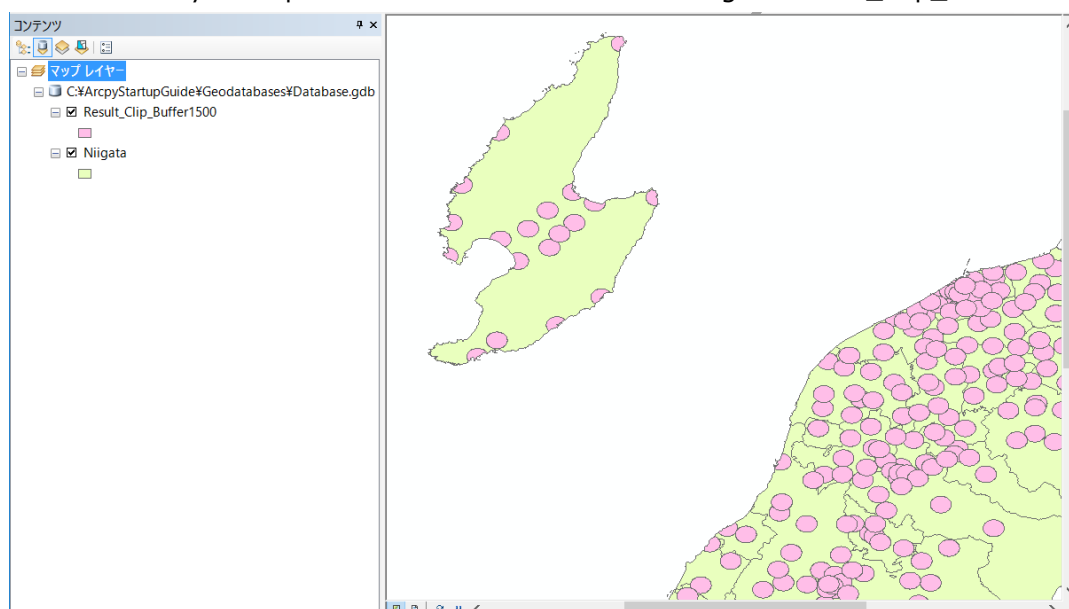
Python モジュールで作成したスクリプトを保存します。

- [ファイル] メニューから [名前を付けて保存] を選択し、
「C:\¥ArcPyStartupGuide¥Scripts」フォルダーに「Clip_Buffer1500.py」という名前で
[保存] します。



実行したスクリプトにより、「Result_Clip_Buffer1500」ポリゴン フィーチャクラスが作成されていることを確認します。

- ArcMap で [ツール] ツールバー → [新規] ボタンをクリックし、[新規ドキュメント] ダイアログで [OK] をクリックします。
- [ツール] ツールバー → [データの追加] ボタンをクリックし、[データの追加] ダイアログを使用して以下の 2 つのデータをマップに追加します。
 - C:\ArcPyStartupGuide\Geodatabases\Database.gdb\Niigata
 - C:\ArcPyStartupGuide\Geodatabases\Database.gdb\Result_Clip_Buffer1500



[クリップ] ツールにより、新規に「Result_Clip_Buffer1500」が作成されたことが確認できます。沿岸部の海まではみ出ていたポリゴン フィーチャが、クリップによって「Niigata」レイヤーの範囲で切り取られていることがわかります。

- ArcMap の [ファイル] メニューから [名前を付けて保存] を選択し
「C:¥ArcPyStartupGuide¥MapDocuments」フォルダーに、
「PoliceStation_Clip_Buffer1500.mxd」と名前を付けて保存してください。
- ArcMap、PyScripter を終了します。

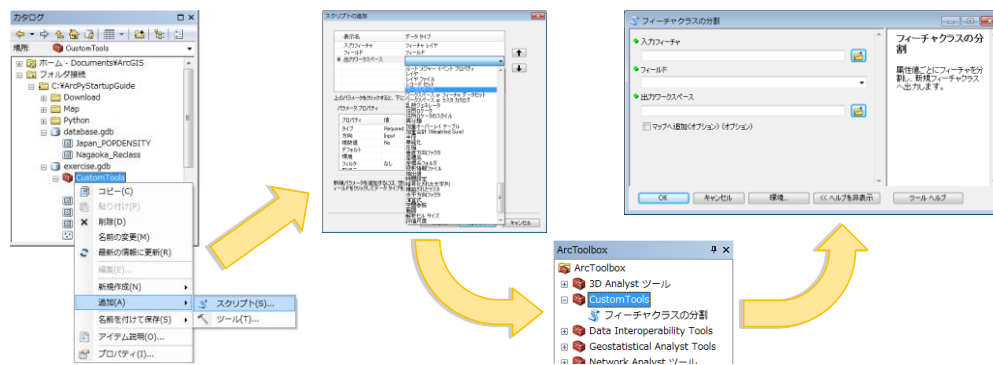


第 3 章 Python を使用した ツールの作成

スクリプト ツールとは

スクリプト ツール

- 作成した Python スクリプトを、ジオプロセシング ツールとして実行できる
 - システム ジオプロセシング ツールと同様に操作
 - 検索ウィンドウやカタログ ウィンドウからも実行可能
- カスタム ツールボックスに作成
 - スクリプト ツールウィザードを通して、ツールとパラメータを定義



スクリプト ツールとは、ユーザー独自の Python スクリプトおよび機能を、ユーザー独自のジオプロセシング ツールに変換することができるツールです。作成したスクリプト ツールはシステム ジオプロセシング ツールと同様の外観を持ち、同じように動作させることができます。

演習 3: スクリプト ツールの作成

この演習では、Python スクリプトを作成して、オリジナルのジオプロセッシング ツールを作成します。独自に作成した Python スクリプトは、ArcToolbox のツールとして追加することで、ArcToolbox にある他のジオプロセッシング ツール（ディゾルブやバッファーなど）と同様に利用することができます。

この演習では、フィーチャクラスの属性値を使用して、フィールド内で同じ属性値を持つフィーチャを 1 つの固まりとして別のフィーチャクラスへ新規に出力するスクリプト ツールを作成します。



演習時間：20 分

ステップ 1: スクリプト ファイルの作成

このステップでは、PyScripter を使用してカスタム ツールボックスに追加するスクリプトを作成します。

- PyScripter を起動します（デスクトップのショートカットアイコン、または Windows の [スタート] プログラム → [PyScripter] → [PyScripter for Python 2.7]）。
- [ファイル] メニューから [名前を付けて保存] を選択し、
「C:\ArcPyStartupGuide\Scripts」フォルダーに「SplitFeatureClassTool.py」という名前でスクリプトを保存します。
- PyScripter の Python モジュールに以下のコードを記入します。

```
# coding:cp932
#
#-----
# Name:          SplitFeatureClassTool.py
# Purpose:       属性値ごとにフィーチャクラスを分割します。
#
# Author:
#
# Created:       <今日の日付>
#-----
```

- ArcPy サイトパッケージと os モジュールをインポートします。

```
import arcpy, os
```

- GetParameterAsText 関数の戻り値を変数に代入します。この関数は、ツール実行時に指示されるパラメーターの値が入ります。

```
inputFL = arcpy.GetParameterAsText(0)
field = arcpy.GetParameterAsText(1)
outputws = arcpy.GetParameterAsText(2)
addMap = arcpy.GetParameterAsText(3)
```

- Describe 関数を使用して、ワークスペースのプロパティ情報を取得します。

```
desc = arcpy.Describe(outputws)
```

- da.SearchCursor 関数を使用して、分割するデータの Cursor を取得します。

```
cur = arcpy.da.SearchCursor(inputFL, field)
```

- 空の辞書型オブジェクトを作成します。

```
dic = {}
```

- Cursor 内の各 row オブジェクト (レコード) に for 文内のコードを実行します。

```
for row in cur:
    dic[row[0]] = "a"
```

- データへの参照を解放します。

```
del cur, row
```

- 辞書型の keys 関数でオブジェクトが持つキーをリスト型で取得できるのを利用し、リスト内の属性値を使って、[選択] ツールを実行します。

```
i = 0
for fc in dic.keys():
    if desc.workspaceType == "FileSystem":
        output = outputws + os.sep + fc + ".shp"
    elif desc.workspaceType == "LocalDatabase":
        output = outputws + os.sep + fc
    exp = '¥' + field + '¥' + "=¥" + fc + "¥"
    out = arcpy.Select_analysis(inputFL, output, exp)
    arcpy.AddMessage(fc)
    i += 1
```

- 作成したデータをマップに追加します。

```
if addMap == "true":
    mxd = arcpy.mapping.MapDocument("CURRENT")
    df = mxd.activeDataFrame
    lyr = arcpy.MakeFeatureLayer_management(out, fc).getOutput(0)
    arcpy.mapping.AddLayer(df, lyr)
```

- 描画の更新を行います。

```
arcpy.RefreshActiveView()
arcpy.RefreshTOC()
del mxd, lyr
```

- メッセージを追加し、変数を解放します。

```
arcpy.AddMessage(outputws + u" に " + str(i) + u" フィーチャクラスの出力が完了しました。")
del desc, outputws
```



Python では、インデントも言語を構成する一部なので、同じスペース数でインデントを付けてください。

- 記述したスクリプトを上書き保存し、PyScripter を終了します。

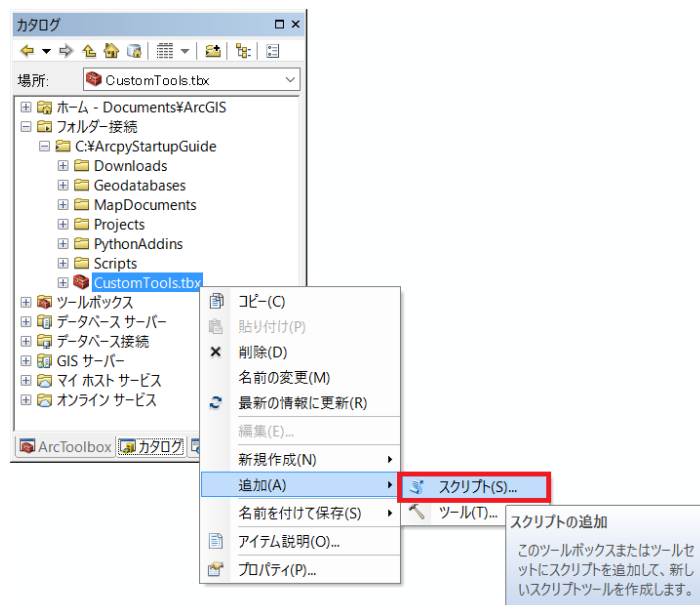
ステップ 2: スクリプト ツールの作成

このステップでは、作成したスクリプトをカスタム ツールボックスに追加して、スクリプト ツールを作成します。

- ArcMap を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcMap 10.7])。
- [ArcMap - はじめに] ダイアログが表示されたら [キャンセル] をクリックします。
- [カタログ] ウィンドウの「C:\ArcPyStartupGuide」フォルダー上で右クリックし、[新規作成] → [ツールボックス] を選択します。
- ツールボックスの名前を「CustomTools.tbx」に変更します。
- 「CustomTools.tbx」を右クリックし、[プロパティ] を選択します。
- [CustomTools プロパティ] ダイアログの [一般] タブで [エイリアス] に「custom」と入力します。
- [OK] をクリックし、[CustomTools プロパティ] ダイアログを閉じます。

ステップ 1 で作成したスクリプトを、「CustomTools.tbx」ツールボックス内に追加します。

- 「CustomTools.tbx」ツールボックスを右クリックし、[追加] → [スクリプト] を選択します。



- [スクリプトの追加] ダイアログで、以下の内容を入力し [次へ] をクリックします。

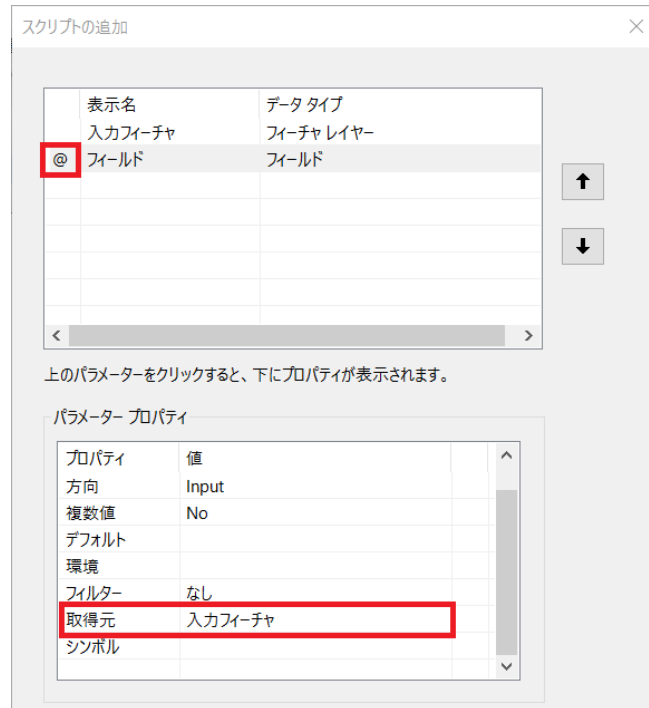
- 名前 : 「SplitFeatureClassTool」と入力
- ラベル : 「フィーチャクラスの分割」と入力
- 説明 : 「属性値ごとにフィーチャを分割し、
新規フィーチャクラスへ出力します。」と入力
- スタイルシート : 記入なし
- 相対パスで保存 : チェックあり
- 常にフォアグラウンドで実行 : チェックあり

- [スクリプト ファイル] に「C:¥ArcPyStartupGuide¥Scripts¥SplitFeatureClassTool.py」スクリプト ファイルを指定し、その他の項目はそのままで [次へ] をクリックします。

次に、4 つのスクリプト引数や引数のプロパティについての設定をします。

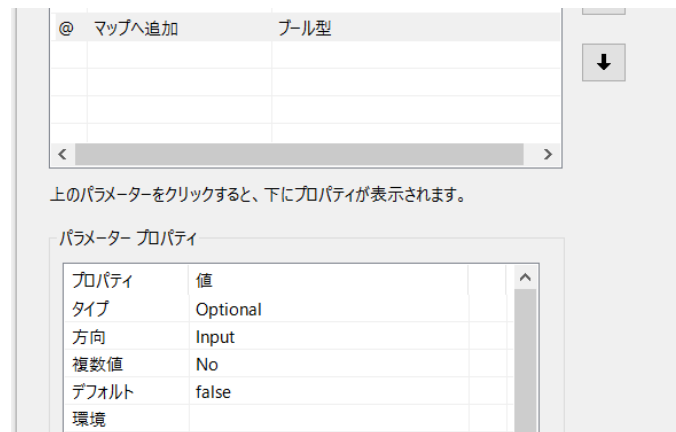
- [表示名] に「入力フィーチャ」と入力し、[データ タイプ] を [フィーチャ レイヤー] に設定します。

- [表示名] の「入力フィーチャ」の下に、「フィールド」と入力し、[データ タイプ] を [フィールド] に設定します。
- 「@」記号が、「フィールド」を指している状態で、[パラメーター プロパティ] の [取得元] を [入力フィーチャ] に設定します。



[取得元] を設定しておくことで、[取得元] に設定したパラメーターに入力されるデータを持つテーブル情報を取得して表示します。

- [表示名] の「フィールド」の下に、「出力ワークスペース」と入力し、[データ タイプ] を [ワークスペース] に設定します。
- [表示名] の「出力ワークスペース」の下に、「マップへ追加」と入力し、[データ タイプ] を [ブール型] に設定します。
- 「@」記号が、「マップへ追加」を指している状態で、[パラメーター プロパティ] を以下に設定します。
 - タイプ : [Optional] に設定
 - デフォルト : [False] に設定

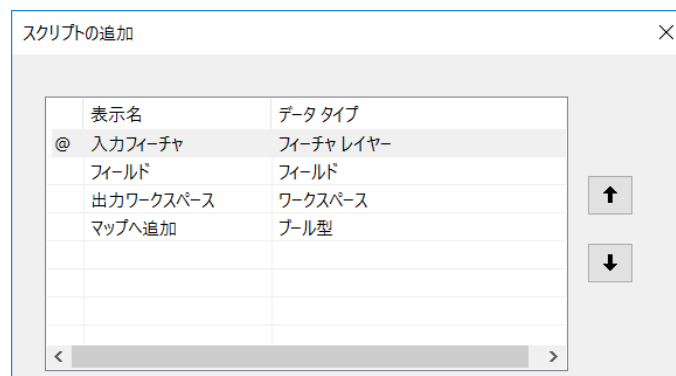


[パラメーター プロパティ] の [タイプ] を、[Required] から [Optional] に変更し、オプションパラメーターに設定すると、パラメーターの入力をしなくてもツールの実行ができます。



[False] を選択すると、[デフォルト] には [false] と入力されます。

- [スクリプトの追加] ダイアログが下図と一致しているか確認します。

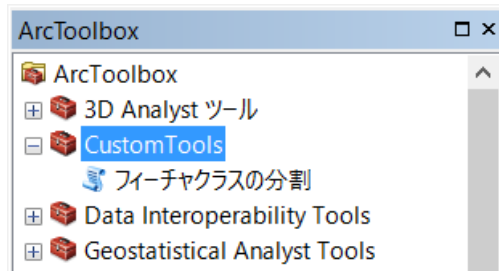


- [完了] をクリックします。

スクリプト ツールが完成しました。

作成したスクリプト ツールを含むカスタム ツールボックスを、[ArcToolbox] ウィンドウに追加することができます。カスタム ツールボックスを追加することで、すばやくカスタム ツールにアクセスできます。

- [ArcToolbox] ウィンドウを表示し、[ArcToolbox] を右クリックして [ツール ボックスの追加] を選択します。
- 「C:\ArcPyStartupGuide\CustomTools.tbx」を選択し、[開く] をクリックします。

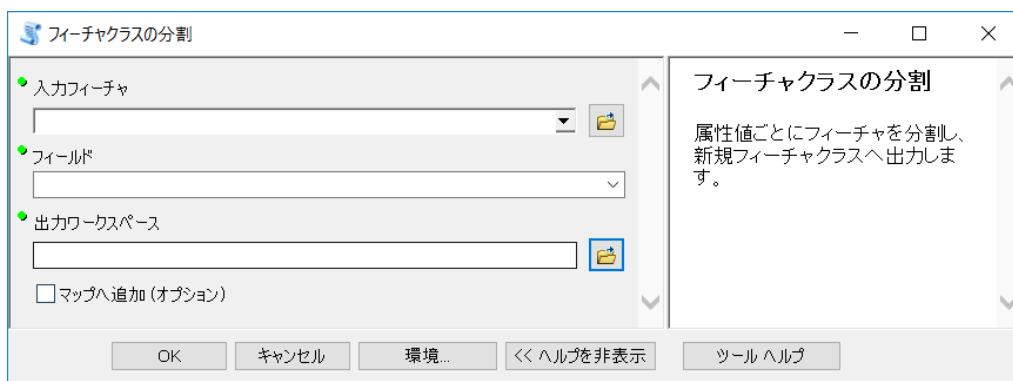


[ArcToolbox] に作成した [CustomTools] ツールボックスが追加されました。

ステップ 3: スクリプト ツールの実行

このステップでは、ステップ 2 で作成した「フィーチャクラスの分割」ツールを実行します。演習 2B で作成した「Nagaoka_Reclass」フィーチャクラスを使用して、[Reclass] フィールドの属性値ごとに新規にレイヤーを出力します。演習 2B を行われていない方は、任意のデータを使用してツールを実行してください。

- [ツール] ツールバー → [データの追加] ボタンをクリックし、[データの追加] ダイアログを使用して以下のデータをマップに追加します。
 - C:\¥ArcPyStartupGuide¥Geodatabases¥Exercise.gdb¥Nagaoka_Reclass
- [ArcToolbox] ウィンドウの [CustomTools] → [フィーチャクラスの分割] ツールをダブルクリックします。



「フィーチャクラスの分割」ツールでは、[スクリプトの追加] ダイアログで設定した 4 つのパラメーター（入力フィーチャクラス、フィールド、出力ワークスペース、マップへ追加）を入力するように指示しています。

[スクリプトの追加] ダイアログで記述したツールの説明文がヘルプ パネルに表示されます。



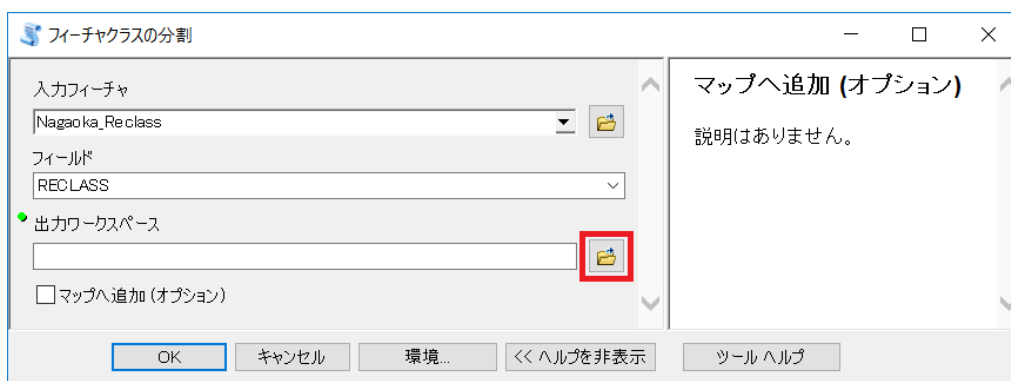
[フィーチャクラスの分割] ダイアログの右側にヘルプが表示されていない場合は、[ヘルプを表示>>] をクリックします。

- [フィーチャクラスの分割] ダイアログの、[入力フィーチャ] のドロップダウン リストか

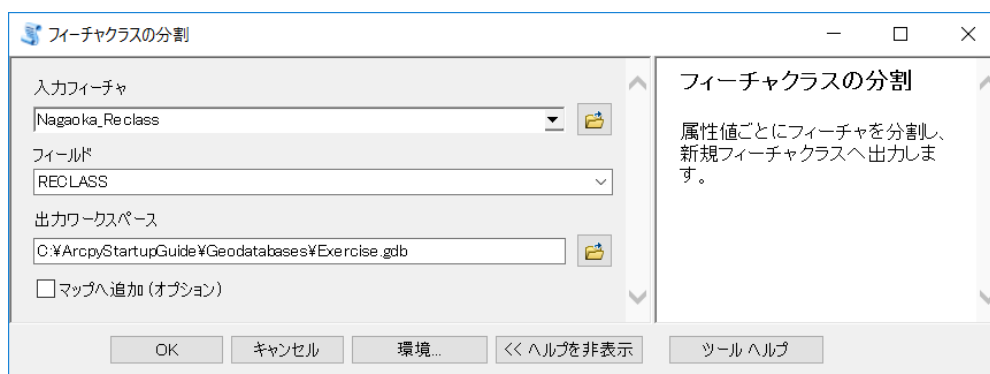
ら「Nagaoka_Reclass」レイヤーを選択します。

[フィールド] のドロップダウン リストを開くと、「Nagaoka_Reclass」レイヤーのフィールドの一覧が表示されます。これは、[スクリプトの追加] ダイアログの設定で、[フィールド] の [パラメーター プロパティ] の [取得元] を「入力フィーチャ」に選択したため、「Nagaoka_Reclass」レイヤーのフィールド一覧が表示されています。

- フィールド一覧から [RECLASS] を選択します。
- [出力ワークスペース] の右側にあるフォルダー アイコンをクリックします。

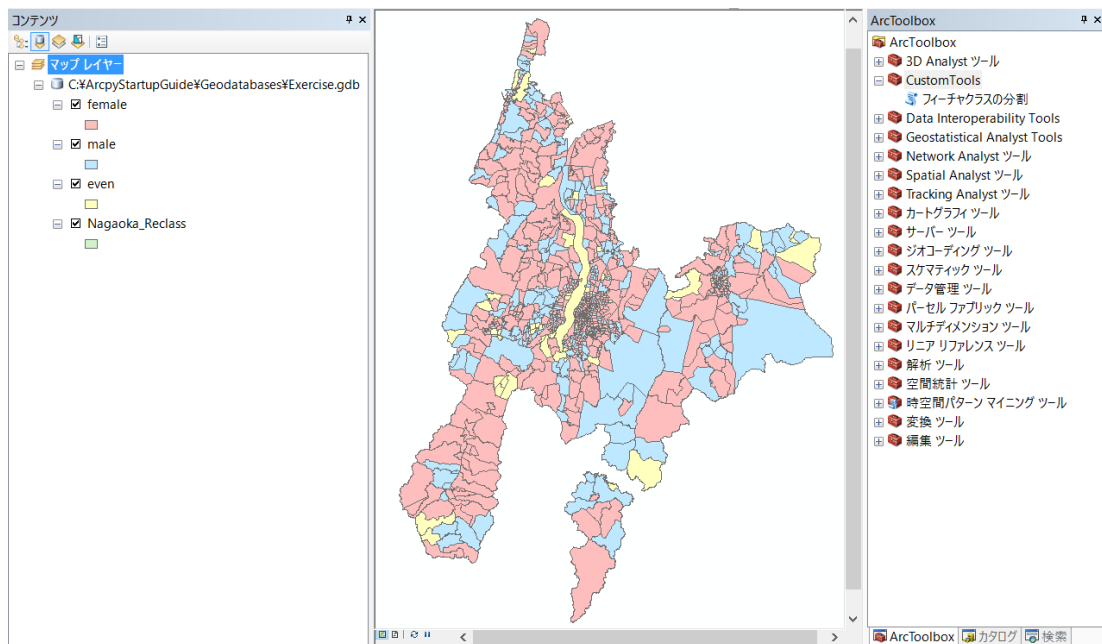


- [出力ワークスペース] ダイアログで
「C:¥ArcPyStartupGuide¥Geodatabases¥Exercise.gdb」を選択し、[追加] をクリック
します。
- [マップへ追加 (オプション)] にチェックを入れます。
- [OK] をクリックし、ツールを実行します。



属性値の種類が多いフィールドを選択する場合は、ツールの実行処理に時間がかかる可能性があります。

- [フィーチャクラスの分割] ダイアログに「完了」のメッセージが表示されたら [閉じる] を
クリックし、ダイアログを閉じます。



ArcMap にツールの実行結果として、「female」「male」「even」レイヤーが新規に追加されました。

以上のステップで、作成したスクリプト ツールを ArcToolbox に追加し、ジオプロセシングツールとして実行することができました。

- [ファイル] メニューから [名前を付けて保存] を選択し

「C:¥ArcPyStartupGuide¥MapDocuments」フォルダーに「SplitFeatureClassTool.mxd」と名前を付けて保存してください。

- ArcMap を終了します。

第 4 章 Python アドインの 作成



アドインの作成方法

ArcGIS 10.1 から ArcGIS Desktop アプリケーション用のアドインを Python で作成できるようになりました。

この章では、Python を使用したアドインの作成方法について学習します。

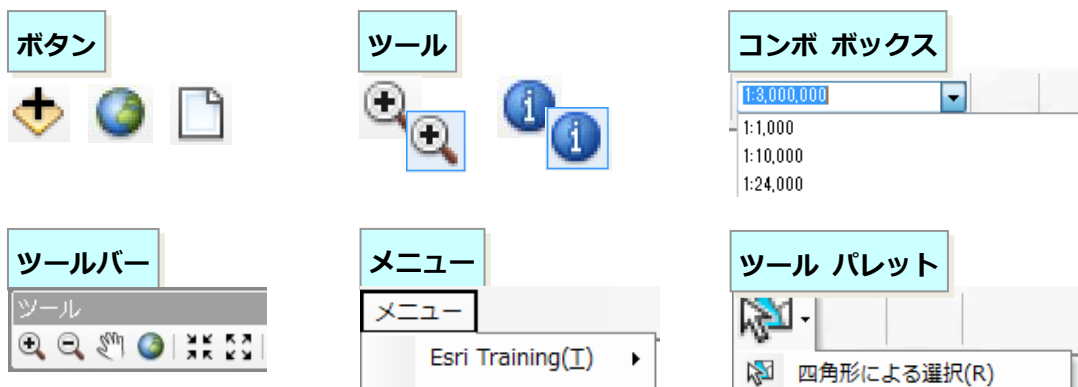
- ✓ アドインとは
- ✓ Python で作成できるアドインの種類

アドインとは

カスタム タスクの実行を補足する機能を提供するために ArcGIS Desktop アプリケーション (ArcMap、ArcCatalog、ArcGlobe、および ArcScene) に組み込む、ツールバー上のツールのコレクションなどのカスタマイズ機能です。

Python で作成できるアドインの種類

- ボタン : 押すと実行されるコマンド
- ツール : 押すと有効化、対話的に操作するコマンド
- コンボ ボックス : ドロップダウン リストから、
選択に基づいて処理するコマンド
- メニュー : ボタン、メニューを格納
- ツール パレット : ツールを格納、関連するツールをグループ化
- ツールバー : 各コマンド、ツール パレット、メニューを
格納
- アプリケーション エクステンション : ArcMap に機能を追加



演習 4A: ボタンの作成

この演習では Python アドイン ウィザードを利用して、ArcMap の ツールバーに追加するボタンを作成する方法を学習します。



演習時間 : 15 分

ステップ 1: プロジェクトの作成

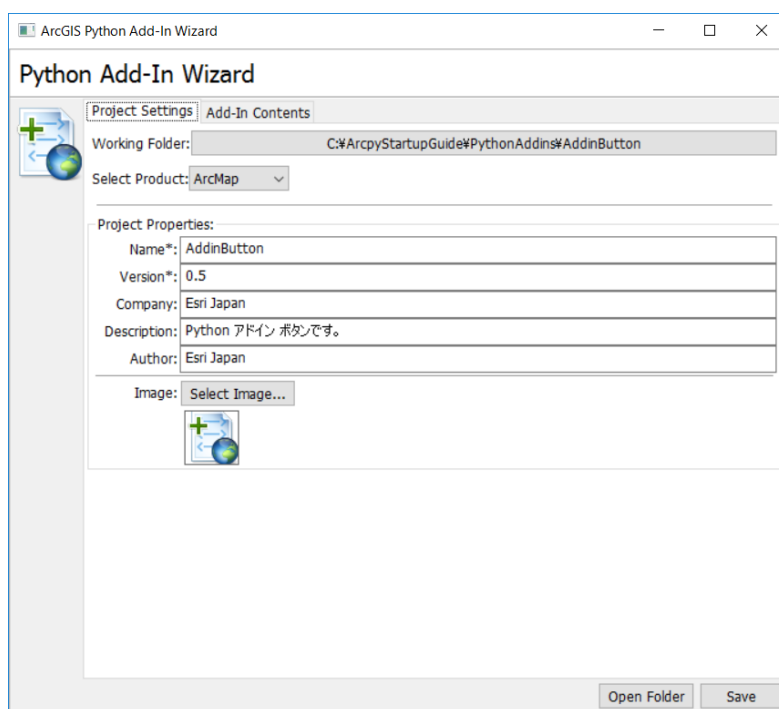
Python アドイン ウィザードから、新規に Python アドイン用のプロジェクトを作成します。

- 演習 1A で任意の場所に保存した [addin_assistant.exe] をダブルクリックします。
- [フォルダー参照] ダイアログで「C:¥ArcPyStartupGuide¥PythonAddin」フォルダーを選択し、[新しいフォルダーの作成] をクリックします。
- 「PythonAddin」フォルダー内に新しいフォルダーが作成されますので、「新しいフォルダー」上で右クリックし、[名前の変更] を選択します。
- フォルダーを「AddinButton」という名前に変更し、[OK] をクリックします。

まず、プロジェクトについて設定を行います。

- [Python Add-In Wizard] ダイアログの [Project Settings] タブで以下の情報を入力します。

- | | | |
|------------------|--------------------------------------|--------------|
| ▪ Select Product | : ArcMap | (対象アプリケーション) |
| ▪ Name | : AddinButton | (ツール名/カテゴリ名) |
| ▪ Version | : 0.5 | (バージョン) |
| ▪ Company | : Esri Japan | (会社/組織) |
| ▪ Description | : Python アドイン ボタンです。 | (説明) |
| ▪ Author | : Esri Japan | (作成者) |
| ▪ Image | : ※デフォルト画像のままで構いません。(アドインに関連づけられた画像) | |



次に、作成するアドインについて設定を行います。まずは、ボタンを格納するためのツールバーを作成します。ボタンは必ずツールバーやメニューに格納されている必要があります。

□ [Python Add-In Wizard] ダイアログの [Add-In Contents] タブをクリックします。

□ [TOOLBARS] 上で右クリックし、[New Toolbar] を選択します。

□ 以下の情報を入力します。

- Caption : Python ツールバー (ボタン) (表示名)
- ID : デフォルトのまま構いません (アドインを識別するための一意の名前)
- Show Initially : チェックあり (アプリケーション起動時に表示/非表示)

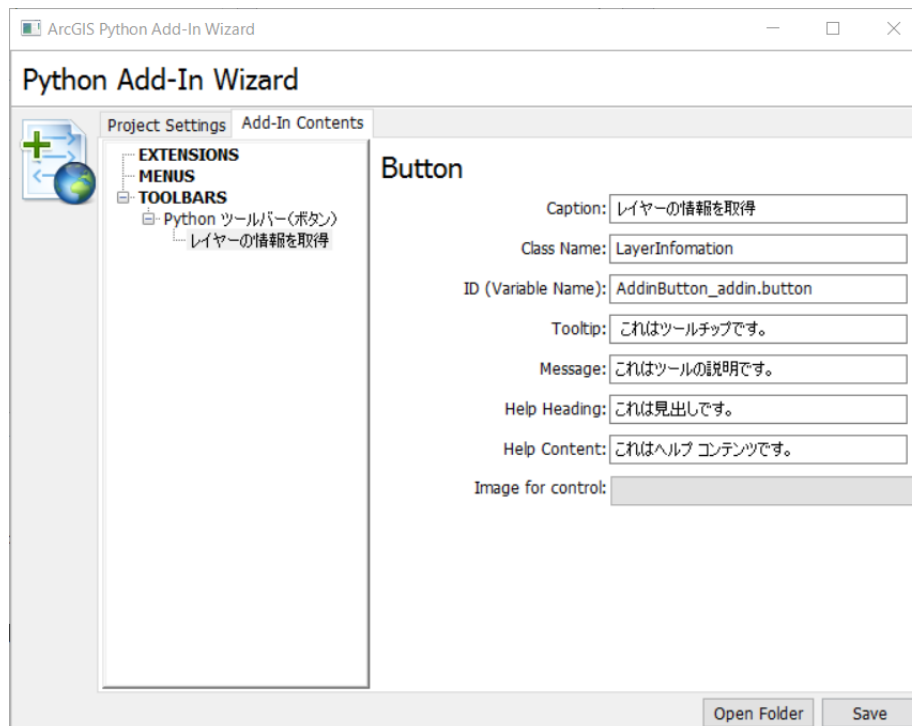
ツールバーが作成できたら、次にツールバーに格納するボタンの作成を行います。

□ 追加された [Python ツールバー] 上で右クリックし、[New Button] を選択します。

□ 以下の情報を入力します。

- Caption : レイヤーの情報を取得 (ボタン名)
- Class Name : LayerInformation (ボタンのクラス名)
- ID : AddinButton_addin.button (アドインを識別するための一意の名前)
- Tooltip : これはツールチップです。 (ツールチップ)
- Message : これはツールの説明です。 (説明)
- Help Headling : これは見出しです。 (ヘルプ コンテンツの見出し)

- Help Content : これはヘルプ コンテンツです。 (ヘルプ コンテンツ)
- Image for control: ※選択しない場合、アイコンは設定されません (アイコン画像)



□ [Save] ボタンをクリックします。

[Save] ボタンをクリックした時点で、プロジェクト フォルダー内に設定したツールバーやボタンを作成するためのファイル群が作成されます。

ステップ 2: コードの記述

次に、ボタンをクリックした際の動作をコードで記述していきます。この演習では、ArcMap の [コンテンツ] ウィンドウで選択しているレイヤーの情報を、メッセージ ボックスで表示させるボタンを作成します。

- [ArcGIS Python Add-In Wizard] ダイアログの [Open Folder] ボタンをクリックします。
- 表示されたプロジェクト フォルダーから「Install」フォルダーを開きます。
- 「AddinButton_addin.py」ファイルを右クリックし、[プログラムから開く] → [PyScripter] を選択します。

Python アドイン ウィザードでは、作成するアドインの種類に応じた実行処理のブロック (関数定義文) が、自動的に追加された状態でファイルが作成されます。今回はボタンを作成したので、「onClick」(ボタンをクリックした時に動作する) という関数が追加されています。


```

1 import arcpy
2 import pythonaddins
3
4 class LayerInfomation(object):
5     """Implementation for AddinButton_addin.button (Button)"""
6     def __init__(self):
7         self.enabled = True
8         self.checked = False
9     def onClick(self):
10         pass

```

- 文字コードの指定を行うコメント文を 1 行目に記述します。

```
# coding: cp932
```

- 「onClick(self)」ブロック内を以下のコードに変更します。

```

def onClick(self):
    mxd = arcpy.mapping.MapDocument("Current")
    selectLayer = pythonaddins.GetSelectedTOCLayerOrDataFrame()
    desc = arcpy.Describe(selectLayer)
    layerName = desc.name
    shpType = desc.shapeType
    featureCount = arcpy.GetCount_management(selectLayer)
    spRef = desc.spatialReference.name
    filePath = desc.path

    pythonaddins.MessageBox(u"レイヤー名 : " + layerName + "%n" +
                             u"シェープタイプ : " + shpType + "%n" +
                             u"フィーチャ数 : " + str(featureCount) + "%n" +
                             u"座標系 : " + spRef + "%n" +
                             u"ファイルパス : " + filePath, u"レイヤー情報")

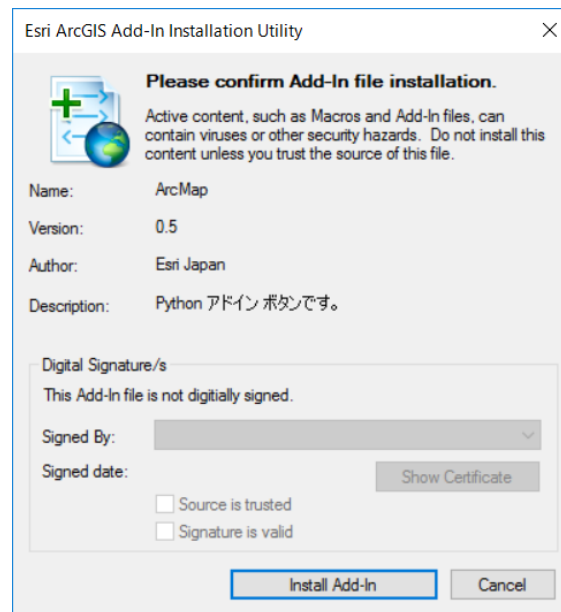
```

- [実行] メニューの [構文チェック] を選択し、構文エラーが発生する場合は修正します。
- 「AddinButton_addin.py」ファイルを上書き保存し、PyScripter を閉じます。
- [ArcGIS Python Add-In Wizard] ダイアログを閉じます。

ステップ 3: 動作のテスト

アドイン ボタンを作成し、ArcMap 上で作成したボタンの動作をテストします

- 下記フォルダー内にある「makeaddin.py」をダブルクリックします。
- C:\ArcpyStartupGuide\PythonAddins\AddinButton
- 先ほど Python アドイン ウィザードで行った設定を基に自動でプロジェクト フォルダー内に「AddinButton.esriaddin」アドイン ファイルが作成されます。
- 「AddinButton.esriaddin」アドイン ファイルをダブルクリックします。
- [Esri ArcGIS Add-In Installation Utility] ダイアログの [Install Add-In] をクリックします。



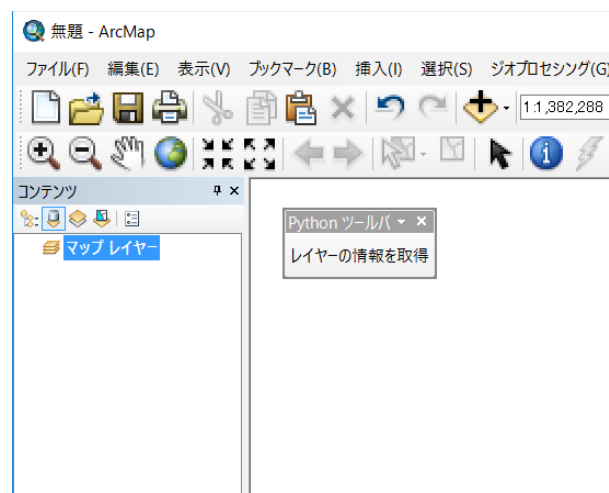
- 「Installation succeeded.」というメッセージ ボックスが表示されたら [OK] をクリックします。

これで ArcGIS に Python アドインがインストールされました。作成したアドインを他のユーザーと共有する場合も、「AddinButton.esriaddin」アドイン ファイルを他のユーザーに渡し、[Esri ArcGIS Add-In Installation Utility] ダイアログから [Install Add-In] を選択するだけで可能です。

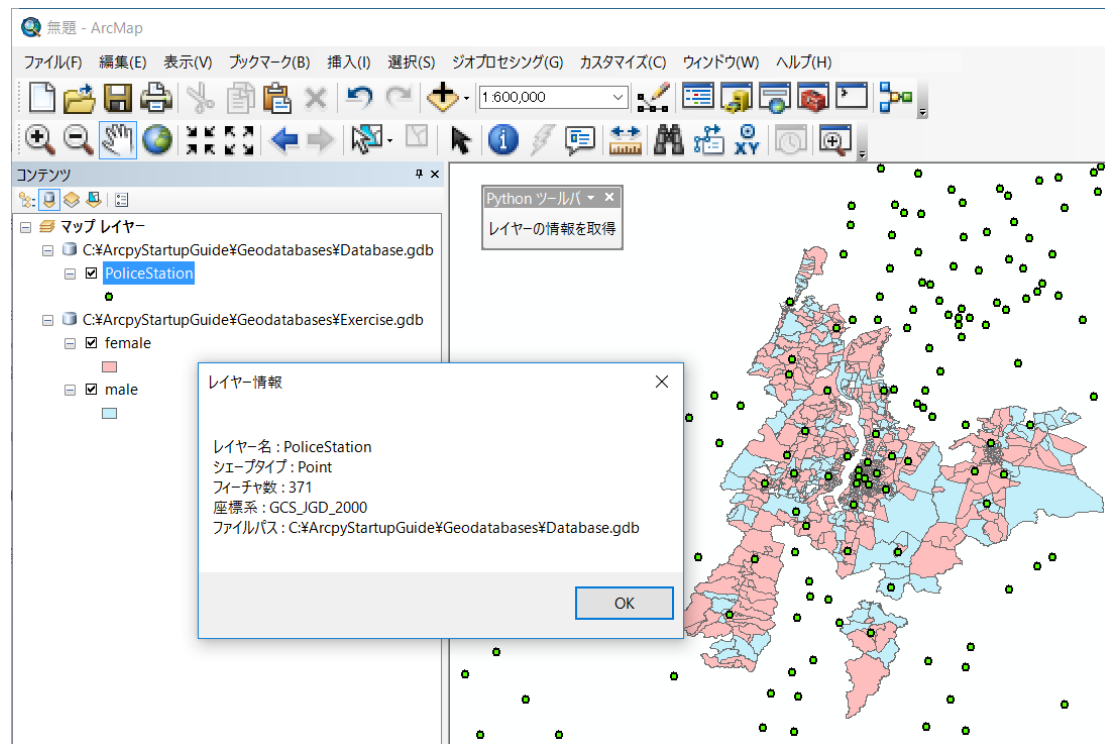
次に、インストールしたアドインの動作を確認します。

- ArcMap を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcMap 10.7])。
- [ArcMap - はじめに] ダイアログが表示されたら [キャンセル] をクリックします。

[ArcGIS Python Add-In Wizard] でツールバーを作成する際に、[Show Initially] ヘチェックをしたため、ArcMap の起動時に自動で作成したツールバーが追加されています。



- [ツール] ツールバーから [データの追加] ボタンをクリックし、[データの追加] ダイアログを使用して以下のデータをマップに追加します。
 - C:\ArcPyStartupGuide\Geodatabases\Exercise.gdb\male
 - C:\ArcPyStartupGuide\Geodatabases\Exercise.gdb\female
 - C:\ArcPyStartupGuide\Geodatabases\Database.gdb\PoliceStation
- ArcMap の [コンテンツ] ウィンドウに追加したレイヤーを選択し、「レイヤーの情報を取得」ボタンをクリックします。



メッセージ ボックスに、選択している「レイヤー名」、「シェープ タイプ」、「フィーチャ数」、「座標系」、「ファイルパス」情報が表示されることを確認します。他のレイヤーでも「選択レイヤーの情報を取得」ボタンをクリックして試してみましょう。

- ArcMap の [ファイル] メニューから [名前を付けて保存] を選択し
「C:\ArcPyStartupGuide\MapDocuments」フォルダーに「PythonAddin_Button.mxd」と名前を付けて保存してください。
- ArcMap を終了します。

演習 4B: ツールの作成

この演習では Python アドイン ウィザードを利用して、ArcMap の ツールバーに追加するツールを作成する方法を学習します。



演習時間 : 15 分

ステップ 1: プロジェクトの作成

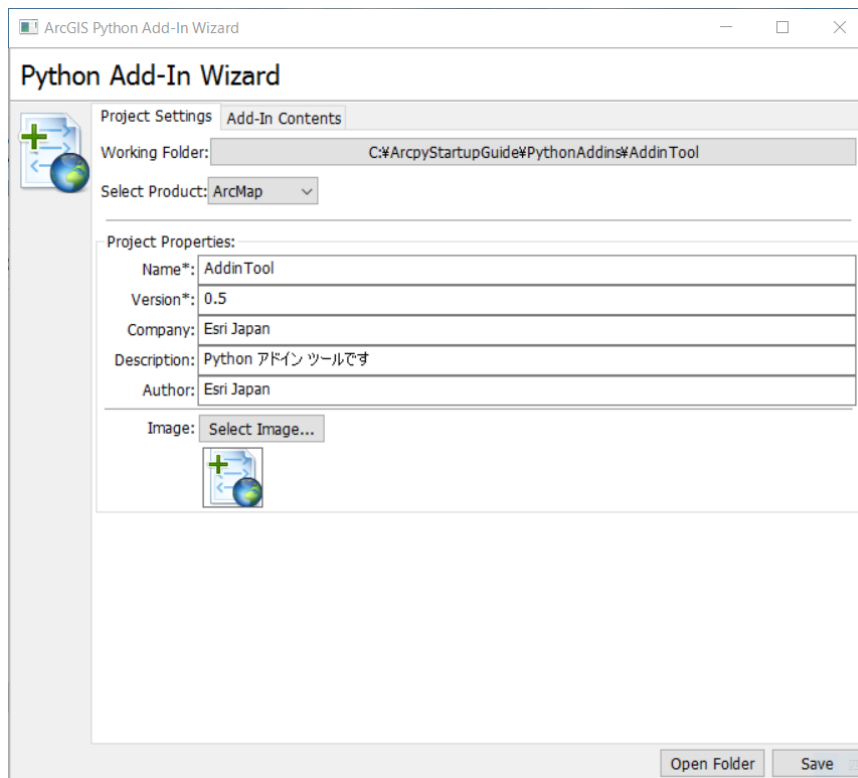
Python アドイン ウィザードから、新規に Python アドイン用のプロジェクトを作成します。

- 演習 1A で任意の場所に保存した [addin_assistant.exe] をダブルクリックします。
- [フォルダー参照] ダイアログで「C:¥ArcPyStartupGuide¥PythonAddin」フォルダーを選択し、[新しいフォルダーの作成] をクリックします。
- 「PythonAddin」フォルダー内に新しいフォルダーが作成されますので、「新しいフォルダー」上で右クリックし、[名前の変更] を選択します。
- フォルダーを「AddinTool」という名前に変更し、[OK] をクリックします。

まず、プロジェクトについて設定を行います。

- [Python Add-In Wizard] ダイアログの [Project Settings] タブで以下の情報を入力します。

- | | | |
|------------------|--------------------------------------|--------------|
| ▪ Select Product | : ArcMap | (対象アプリケーション) |
| ▪ Name | : AddinTool | (ツール名/カテゴリ名) |
| ▪ Version | : 0.5 | (バージョン) |
| ▪ Company | : Esri Japan | (会社/組織) |
| ▪ Description | : Python アドイン ツールです。 | (説明) |
| ▪ Author | : Esri Japan | (作成者) |
| ▪ Image | : ※デフォルト画像のままで構いません。(アドインに関連づけられた画像) | |



次に、作成するアドインについて設定を行います。まずは、ツールを格納するためのツールバーを作成します。ツールは必ずツールバーやメニューにツールパレット格納されている必要があります。

□ [Add-In Contents] タブをクリックします。

□ [TOOLBARS] 上で右クリックし、[New Toolbar] を選択します。

□ 以下の情報を入力します。

- Caption : Python ツールバー (ツール) (表示名)
- ID : デフォルトのまま構いません (アドインを識別するための一意の名前)
- Show Initially : チェックなし (アプリケーション起動時に表示/非表示)

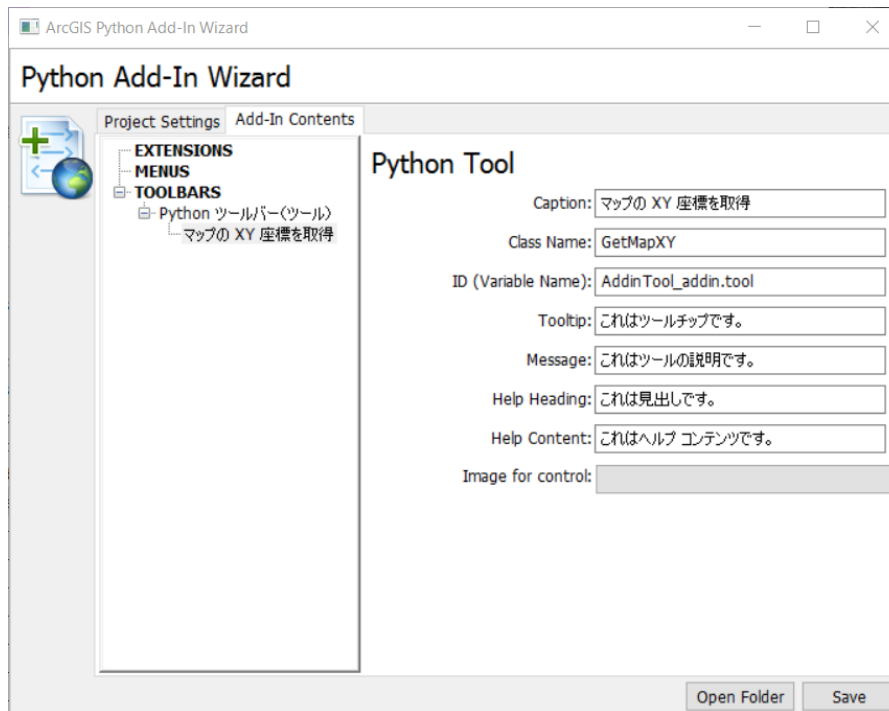
ツールバーが作成できたら、次にツールバーに格納するボタンの作成を行います。

□ 追加された [Python ツールバー (ツール)] 上で右クリックし、[New Tool] を選択します。

□ 以下の情報を入力します。

- Caption : マップの XY 座標を取得 (ツール名)
- Class Name : GetMapXY (ツールのクラス名)
- ID : AddinTool_addin.tool (アドインを識別するための一意の名前)

- Tooltip : これはツールチップです。 (ツールチップ)
- Message : これはツールの説明です。 (説明)
- Help Headling : これは見出しです。 (ヘルプ コンテンツの見出し)
- Help Content : これはヘルプ コンテンツです。 (ヘルプ コンテンツ)
- Image for control : ※選択しない場合、アイコンは設定されません (アイコン画像)



- [Save] ボタンをクリックします。

[Save] ボタンをクリックした時点で、プロジェクト フォルダー内に設定したツールバーやツールを作成するためのファイル群が作成されます。

ステップ 2: コードの記述

次に、ツールをクリックした際の動作をコードで記述していきます。この演習では、ArcMap のマップ上でクリックした際に、クリックした箇所の座標値をメッセージボックスで表示するツールを作成します。

- [ArcGIS Python Add-In Wizard] ダイアログの [Open Folder] ボタンをクリックします。

表示されたプロジェクト フォルダーから「Install」フォルダーを開きます。

- 「AddinTool_addin.py」ファイルを右クリックし、[プログラムから開く] → [PyScripter] を選択します。

Python アドイン ウィザードでは、作成するアドインの種類に応じた実行処理のブロック (関数定義文) が、自動的に追加された状態でファイルが作成されます。今回はツールを作成し

たので、「onMouseDownMap」（マップ上でマウスが押下された場合）、「onKeyDown」（キーボードのキーが押下された場合）といったツールの動作に呼応した関数が追加されています。

```

1 import arcpy
2 import pythonaddins
3
4 class GetMapXY(object):
5     """Implementation for AddinTool_addin.tool (Tool)"""
6     def __init__(self):
7         self.enabled = True
8         self.shape = "NONE" # Can set to "Line", "Circle" or "Rectangle" fori
9     def onMouseDown(self, x, y, button, shift):
10         pass
11     def onMouseDownMap(self, x, y, button, shift):
12         pass
13     def onMouseUp(self, x, y, button, shift):
14         pass
15     def onMouseUpMap(self, x, y, button, shift):
16         pass
17     def onMouseMove(self, x, y, button, shift):
18         pass
19     def onMouseMoveMap(self, x, y, button, shift):
20         pass
21     def onDblClick(self):
22         pass
23     def onKeyDown(self, keycode, shift):
24         pass
25     def onKeyUp(self, keycode, shift):
26         pass
27     def deactivate(self):
28         pass
29     def onCircle(self, circle_geometry):
30         pass
31     def onLine(self, line_geometry):
32         pass
33     def onRectangle(self, rectangle_geometry):
34         pass

```

- 文字コードの指定を行うコメント文を 1 行目に記述します。

```
# coding:cp932
```

- 「onMouseDownMap (self,x,y,button,shift)」ブロック内を以下のコードに変更します。

```
def onMouseDownMap(self, x, y, button, shift):
    pythonaddins.MessageBox(u"X 座標 : " + str(x) + "\n" +
                             u"Y 座標 : " + str(y), u"クリック箇所の座標値")
```

- [実行] メニューの [構文チェック] を選択し、構文エラーが発生する場合は修正します。
- 「AddinTool_addin.py」ファイルを上書き保存し、PyScripter を閉じます。

ステップ 3: 動作のテスト

アドイン ツールを作成し、ArcMap 上でツールの動作をテストします。

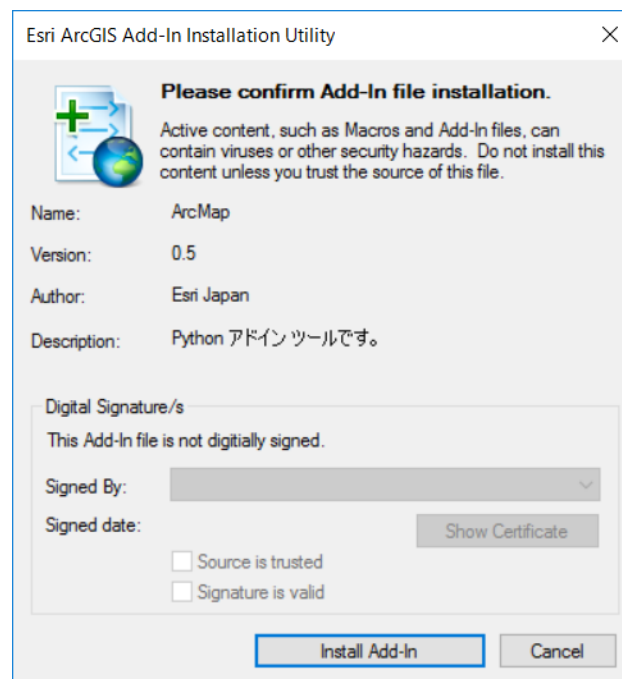
- 下記フォルダー内にある「makeaddin.py」をダブルクリックします。
 - C:\ArcpyStartupGuide\PythonAddins\AddinTool

先ほど Python アドイン ウィザードで行った設定を基に自動でプロジェクト フォルダ内に「AddinButton.esriaddin」アドイン ファイルが作成されます。

- プロジェクト フォルダ内にある「makeaddin.py」をダブルクリックします。

先ほど Python アドイン ウィザードで行った設定を基に自動でプロジェクト フォルダ内に「AddinTool.esriaddin」アドイン ファイルが作成されます。

- 「AddinTool.esriaddin」アドイン ファイルをダブルクリックします。
- [Esri ArcGIS Add-In Installation Utility] ダイアログの [Install Add-In] をクリックします。



- 「Installation succeeded.」というメッセージ ボックスが表示されたら [OK] をクリックします。

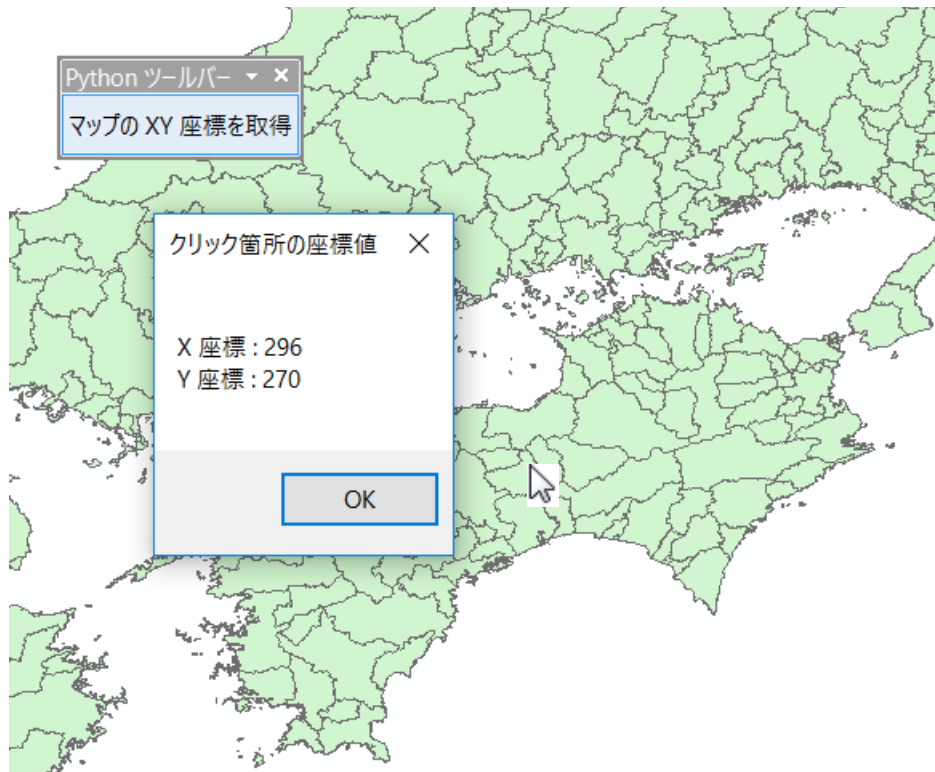
これで ArcGIS に Python アドインがインストールされました。作成したアドインを他のユーザーと共有する場合も、「AddinTool.esriaddin」アドイン ファイルを他のユーザーに渡し、[Esri ArcGIS Add-In Installation Utility] ダイアログから [Install Add-In] を選択するだけで可能です。

次にインストールしたアドインの動作を確認します。

- ArcMap を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcMap 10.7])。
- [ArcMap - はじめに] ダイアログが表示されたら [キャンセル] をクリックします。
- [ツール] ツールバー → [データの追加] ボタンをクリックし、[データの追加] ダイアロ

グを使用して以下のデータをマップに追加します。

- C:¥ArcPyStartupGuide¥Geodatabases¥Database.gdb¥ Japan_ver81
- [カスタマイズ] メニュー → [カスタマイズ モード] → [ツールバー] → [Python ツールバー (ツール)] にチェックを入れ、[閉じる] をクリックします。
- マップ上に追加された [マップの XY 座標を取得] ツールをクリックします。
- マップの任意の箇所をクリックします。



クリックした箇所の座標値がメッセージ ボックスとして表示されます。

- ArcMap の [ファイル] メニューから [名前を付けて保存] を選択し「C:¥ArcPyStartupGuide¥MapDocuments」フォルダーに「PythonAddin_Tool.mxd」と名前を付けて保存してください。
- ArcMap を終了します。



第 5 章 ArcGIS Pro での操作

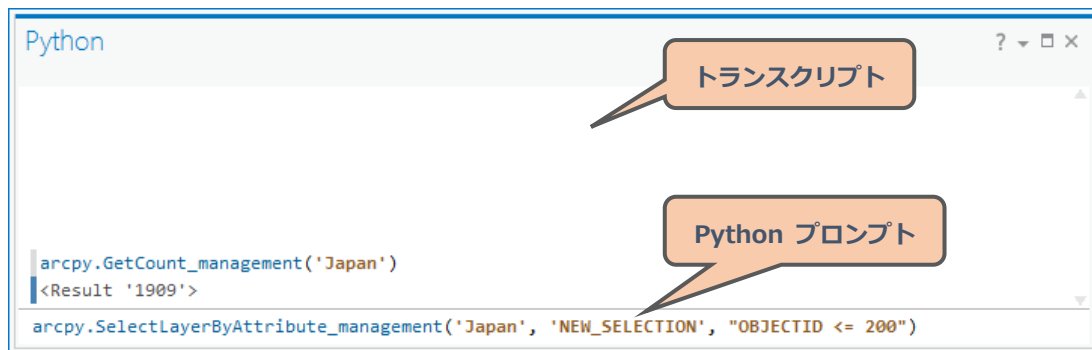
ArcGIS Pro での Python の利用

ArcGIS Pro での Python の使用法は、ArcGIS Desktop、ArcGIS Enterprise、ArcGIS Engine などの他の ArcGIS 製品での使用方法とは異なっています。

- ✓ Python ウィンドウ
- ✓ 外部アプリケーションから Arcpy を実行
- ✓ Arcpy の機能に対する変更
- ✓ Python 3.x へのアップグレード
- ✓ サポートされていないデータ形式

Python ウィンドウ

ArcMap と同様に、ArcGIS Pro 上で、対話的に Python の機能を実行できます。



外部アプリケーションから Arcpy を実行

実行には以下のいずれかの条件を満たしている必要があります。

- ArcGIS Pro へのサイン イン時に「自動的にサイン イン」にチェックしている
- ArcGIS Pro が現在起動している
- ArcGIS Pro をオンライン認証している



上記のいずれかの条件が満たされていない場合は、Python スクリプトの実行時に「RuntimeError」が発生します。



上記の条件は指定ユーザー ライセンス時の条件となるため、SUライセンスで認証されている場合は 3 つの条件に関係なくスクリプトを実行できます。

arcpy の機能に対する変更

- 「arcpy.mapping モジュール」が削除され、「arcpy.mp モジュール」に置き換え

- [カバレッジ] ツールや [Schematics] ツールなど、ArcGIS Pro でサポートされていないツールは実行不可

Python 3.x へのアップグレード

- 文字列が Unicode 型に統一
- print の関数化 (ステートメントとして使うと構文エラー)
- xrange 関数の削除 (range 関数が xrange 関数のように動作)
- dictionary.iteritems メソッドの削除 (dictionary.items メソッドに変換)



Python 2.x で記述したスクリプトに「__future__」モジュールをインポートすると、Python 3.x の新しいルールの一部として実行することができます。



Python のバージョンによる違いの詳細については、第 6 章の補足資料をご参考ください。



Python 3.x では文字コードに標準で UTF-8 が設定されています。

サポートされていないデータ形式

- ジオメトリック ネットワーク
- トポロジ
- パーソナル ジオデータベース
- ラスター カタログ、
- レイヤー / マップ / ジオプロセッシング / タイル マップ パッケージ

演習 5: スクリプト ツールのチェック

ここでは、ArcGIS Pro のジオプロセシング ツールを使用して、第 3 章で作成したスクリプト ツールが ArcGIS Pro で実行可能かチェックし、修正します。



演習時間の目安 : 30分

ステップ 1: プロジェクトの作成

- ArcGIS Pro を起動します (Windows の [スタート] プログラム → [ArcGIS] → [ArcGIS Pro])。
- [ArcGIS サイン イン] ダイアログが表示されたら、[ユーザー名] と [パスワード] を入力してサイン インします。



ArcGIS Online の組織向けアカウントに使用している「ユーザー名」と「パスワード」を入力します。

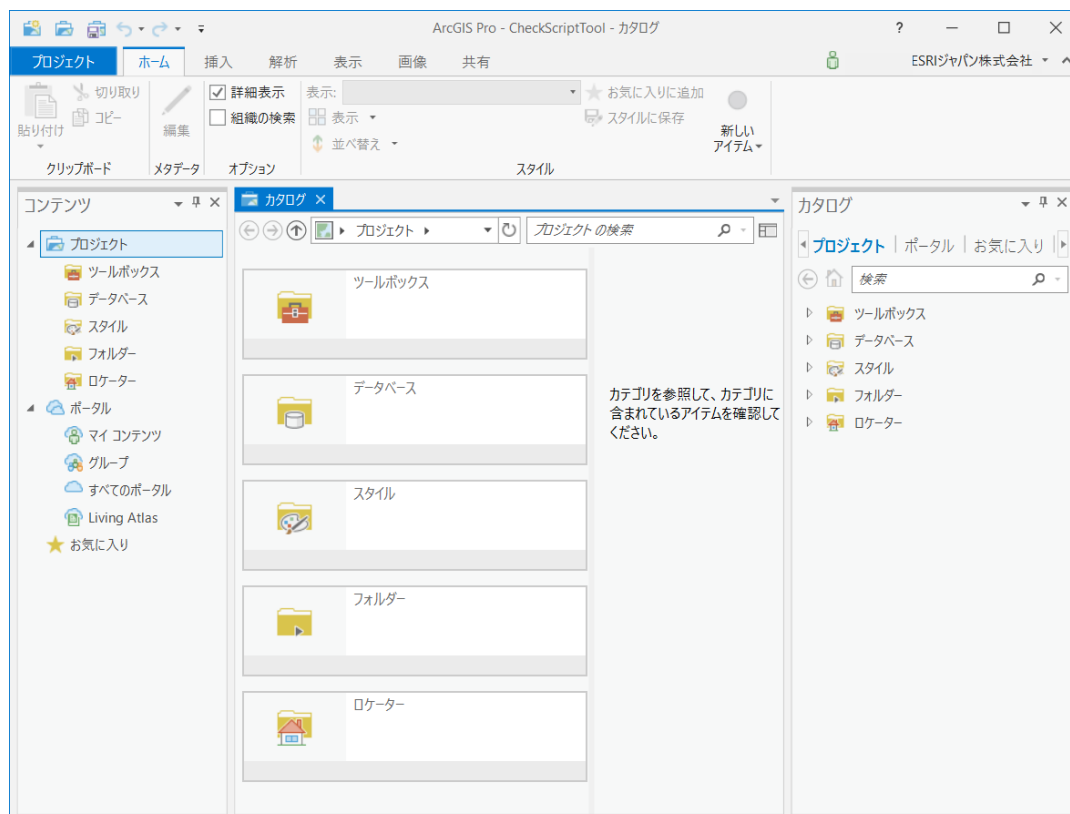
- [ArcGIS Pro] ダイアログが表示されたら、[新しいプロジェクトの作成] から [空のプロジェクト] を選択します。
- [新しいプロジェクトの作成] ダイアログが表示されたら以下を入力して [OK] をクリックします。
 - 名前 : CheckScriptTool
 - 場所 : C:¥ArcPyStartupGuide¥Projects
 - このプロジェクトのための新しいフォルダーを作成 :チェックあり



ArcGIS Pro では「プロジェクト」にマップ、レイアウト、使用するデータやサーバーへの接続、タスクなどを保存します。



プロジェクトはその中に複数のマップを持つことができますので、ArcGIS Desktop のマップ ドキュメント (.mxd) とは異なることに注意してください。



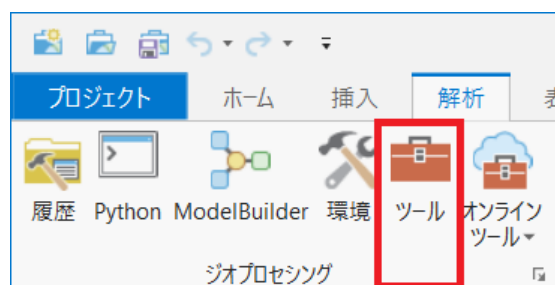
新規に空のプロジェクトが開きます。

ステップ 2: チェック ツールの実行

ArcGIS Pro では、ArcMap で作成されたスクリプト ツールや Python 2.x で作成された Python ファイル (.py) が ArcGIS Pro で実行可能かチェックする [ArcGIS Pro で実行するツールの分析] ツールが登場しました。

このステップでは、[ArcGIS Pro で実行するツールの分析] ツールを使用して、第 3 章で作成したスクリプト ツールが ArcGIS Pro で実行可能かチェックします。

□ [解析] タブ → [ツール] をクリックして [ジオプロセッシング] ウィンドウを表示します。

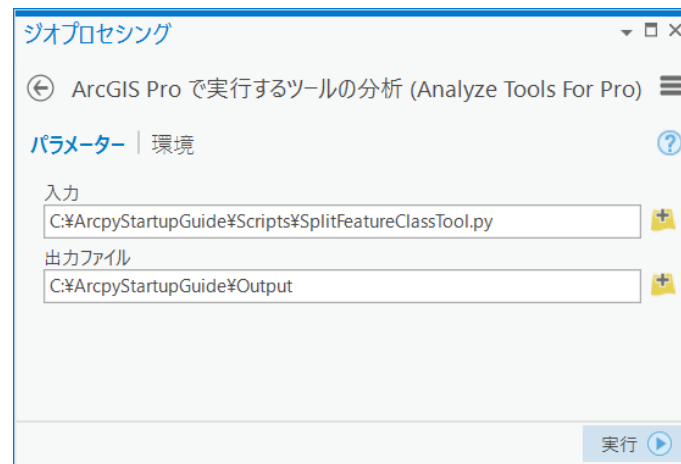


□ [ジオプロセッシング] ウィンドウの [ツールボックス] → [データ管理ツール] → [一般] → [ArcGIS Pro で実行するツールの分析] ツールをクリックします。

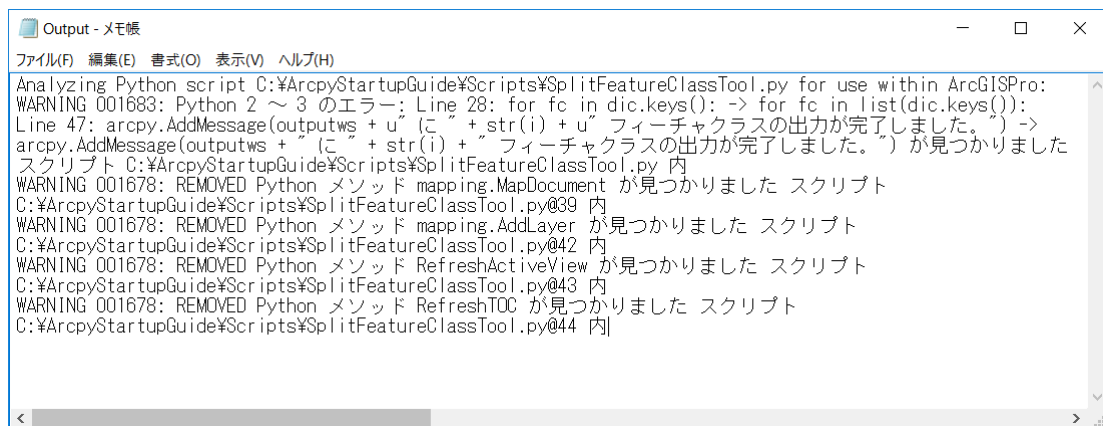


□ [入力] パラメーターに以下を設定して実行します。

- 入力 : C:\ArcPyStartupGuide\Scripts\SplitFeatureClassTool.py
- 出力ファイル : C:\ArcPyStartupGuide\Output.txt



□ 出力された「C:\ArcPyStartupGuide\Output.txt」ファイルを開きます。



「Output.txt」ファイルには、ArcGIS Pro で「SplitFeatureClassTool.py」ファイルを実行するために修正が必要な個所が記されています。「WARNING 001683」は Python 3.x と互換性がないというエラーで、「WARNING 001678」は ArcGIS Pro からは既に削除された ArcPy の機能が見つかったというエラーです。出力された情報を参考にスクリプトを修正します。

- 「C:¥ArcPyStartupGuide¥Scripts」に移動し「SplitFeatureClassTool.py」ファイルをコピーして「C:¥ArcPyStartupGuide¥Scripts」内に貼り付けます。
- 作成された「SplitFeatureClassTool - コピー.py」ファイルの名前を「SplitFeatureClassToolPro.py」に変更します。
- ArcGIS Pro の [カタログ] ウィンドウの [ツールボックス] を右クリックし、[ツールボックスの追加] をクリックします。
- [ツールボックスの追加] ダイアログで「C:¥ArcPyStartupGuide¥CustomTools.tbx」を選択し、[選択] をクリックします。
- [ツールボックス] を展開し、「CustomTools」ツールボックス内の「フィーチャクラスの分割」ツールをコピーして「CustomTools」ツールボックスに貼り付けます。
- 作成された「フィーチャクラスの分割 (2)」ツールを右クリックし、[プロパティ] をクリックします。
- 「スクリプト ツール: フィーチャクラスの分割 (2)」ダイアログの [一般] で設定を以下に変更します。
 - 名前 : SplitFeatureClassToolPro
 - ラベル : フィーチャクラスの分割_Pro
 - スクリプト ファイル :
C:¥ArcPyStartupGuide¥Scripts¥SplitFeatureClassToolPro.py

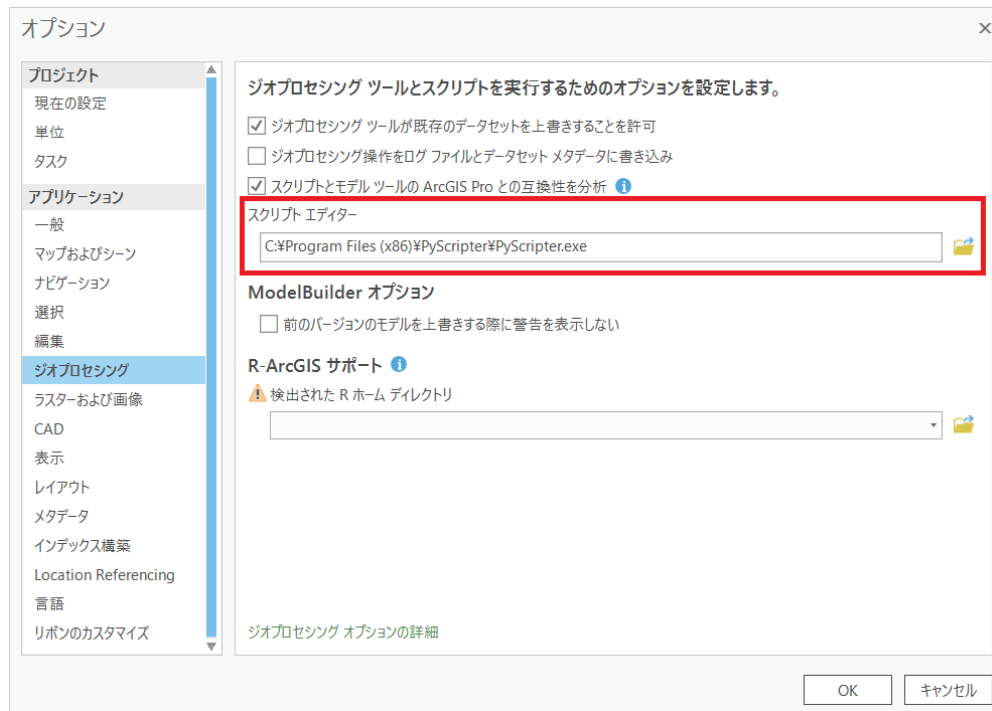


- [OK] をクリックします。

次に、スクリプト ツールまたは Python ツールボックスの編集時にどのアプリケーションまたは Python IDE (統合開発環境) を使用するかを指定します。

- [プロジェクト] タブから [オプション] をクリックします。
- [オプション] ダイアログで [ジオプロセッシング] を選択し、[スクリプトとモデル ツールの ArcGIS Pro との互換性を分析に] チェックをいれます。

- 続けて、[スクリプト エディター] に PyScripter の実行ファイル (PyScripter.exe) を設定し、[OK] をクリックします。



- [カタログ] ウィンドウで「CustomTools」ツールボックスを展開し、「フィーチャクラスの分割_Pro」ツールを右クリックして [編集] をクリックします。

[オプション] で [スクリプト エディター] に指定した PyScripter が起動し、PyScripter 上でスクリプトの編集を行うことができます。

- [エディター ツールバー] ツールバーの [行番号の表示/非表示を切り替え] ツールをクリックします。



- 28 行目を以下に修正します。

```
for fc in list(dic.keys()):
```

- 39 行目を以下に修正します。

```
aprx = arcpy.mp.ArcGISProject("CURRENT")
```



ArcGIS Pro では、マップやレイアウト、ジオデータベースやレイヤーにアクセスする際は、mp モジュールの ArcGISProject クラスから参照していきます。

- 40 行目を以下に修正します。

```
map = aprx.listMaps()[0]
```



プロジェクトに追加したマップは、ArcGISProject クラスの listMaps メソッドから取得します。

- 42 行目を以下に修正します。

```
map.addLayer(lyr)
```



ArcGIS Pro でレイヤー オブジェクトをマップ上に追加するには、mp モジュールの Map クラスの addLayer メソッドを使用します。

- 45 行目を以下に修正します。

```
del aprx, lyr
```

- 47 行目を以下に修正します。

```
arcpy.AddMessage(outputws + " に " + str(i) + " フィーチャクラスの出力が完了しました。")
```



Python 3.x では、文字列が Unicode 型に統一されたので、ArcMap にマルチバイト文字を渡す際に Unicode 型に指定する必要はありません。

- 43、44行目を削除します。



ArcGIS Pro では、強制的にビューを更新する必要はなく、mp モジュールの実行時にアプリケーションが直接更新されます。

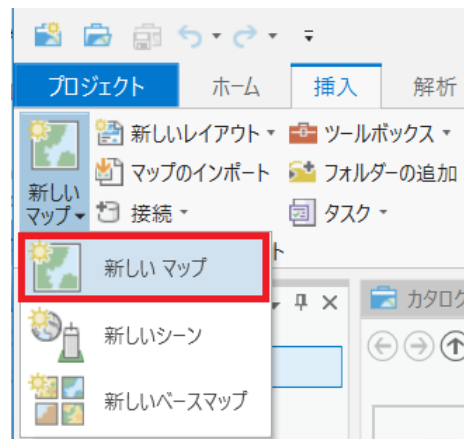
以上でスクリプトの修正が完了しました。

- スクリプトを上書き保存し、Pyscripter を閉じます。

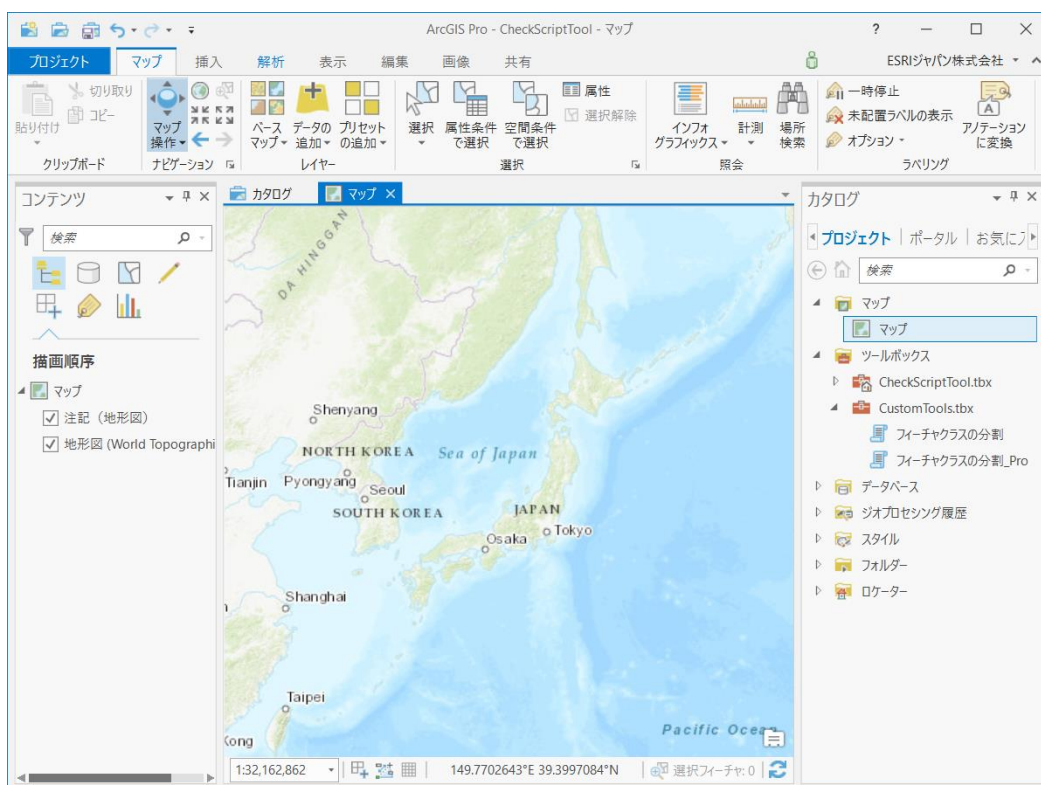
ステップ 3: スクリプト ツールの実行

ArcGIS Pro 上で修正したスクリプト ツールの動作を確認します。

- ArcGIS Pro に戻り、[挿入] タブ → [新しいマップ] のドロップダウン リストから「新しいマップ」をクリックします。

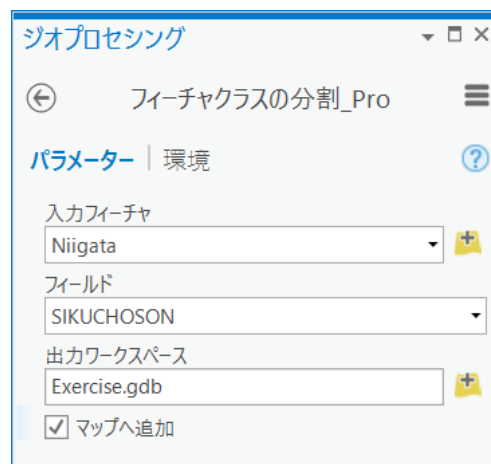


新しいマップを挿入すると、自動的に背景地図が追加されます。



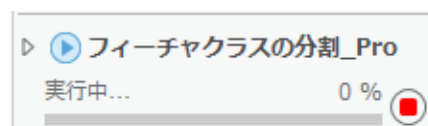
- [カタログ] ウィンドウで [フォルダー] を右クリックし、[フォルダー接続の追加] をクリックします。
- [フォルダー接続の追加] ダイアログで「C:\ArcPyStartupGuide\Geodatabases」を選択し、[OK] をクリックします。
- [カタログ] ウィンドウで [フォルダー] を展開し、[Geodatabases] → [Databases.gdb] → 「Niigata」フィーチャクラスをマップにドラッグ & ドロップで追加します。
- [カタログ] ウィンドウの [ツールボックス] を展開し、[CustomTools.tbx] → [フィーチャクラスの分割_Pro] ツールをダブルクリックします。

- [ジオプロセシング] ウィンドウで、[入力フィーチャ] のドロップダウン リストから「Niigata」レイヤーを選択します。
- [フィールド] のドロップダウン リストから「SIKUCHOSON」を選択します。
- [出力ワークスペース] の右側にあるフォルダー アイコンをクリックします。
- [出力ワークスペース] ダイアログで
「C:\ArcPyStartupGuide\Geodatabases\Exercise.gdb」を選択し、[選択] をクリック
します。
- [マップへ追加] にチェックを入れます。

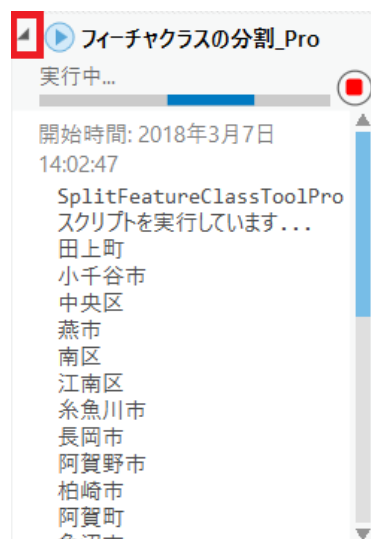


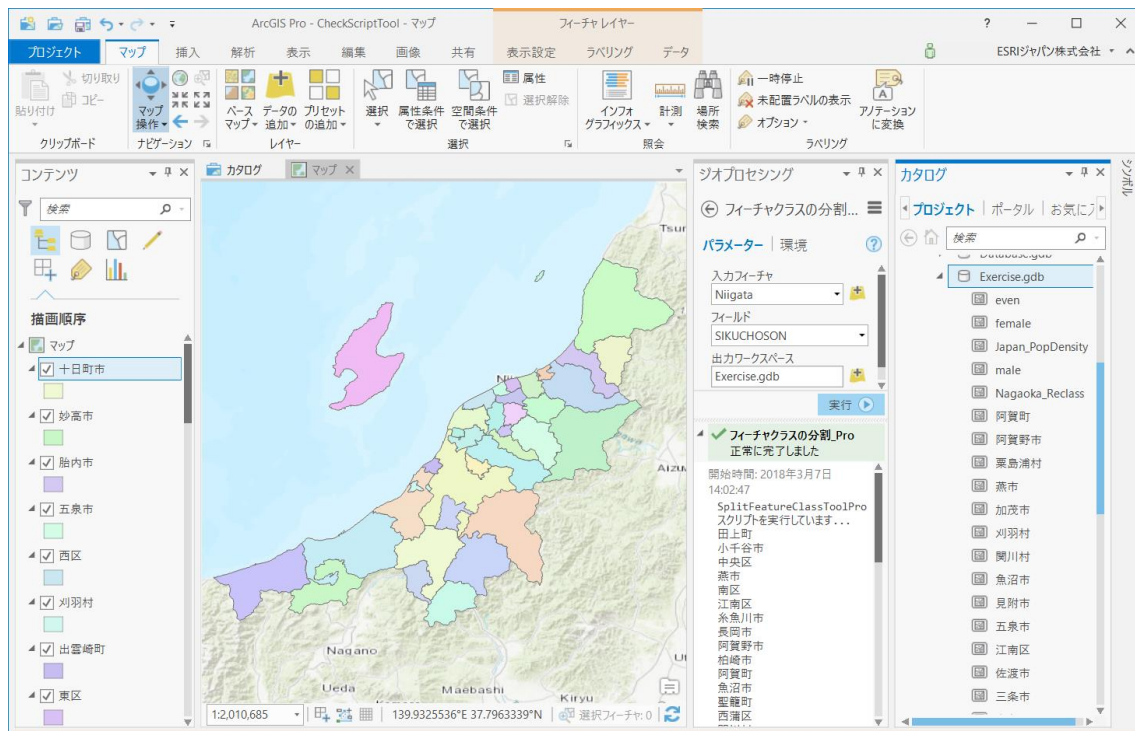
- [実行] をクリックし、ツールを実行します。

[ジオプロセシング] ウィンドウの下部にツールの実行プロセスが表示されます。



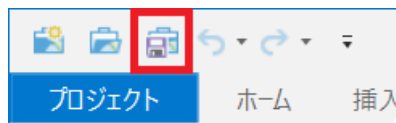
[メッセージの表示/非表示] ボタンをクリックすると、実行メッセージが表示されます。





ツールの実行結果として、新潟県の市区町村のレイヤーが新規に追加されました。ArcMap で作成したスクリプト ツールを修正して、ArcGIS Pro から実行することができました。

□ プロジェクトを上書き保存します。



□ ArcGIS Pro を終了します。



第 6 章 補足資料

参考書籍

- ✓ Python スタートブック (<https://amzn.to/2KfT4eL>)
プログラミング初学者向けに書かれた書籍です。
- ✓ みんなの Python 第 4 版 (<https://amzn.to/2qTqyH4>)
Python 学習の入門用の書籍です。
- ✓ 初めての Python 第 3 版 (<http://amzn.to/1p9cpkq>)
入門から実践まで詳細に書かれた書籍です。

Web サイト

- ✓ 日本 Python ユーザー会 (<http://www.python.jp/>)
日本 Python ユーザー会の Web サイトです。
- ✓ Python 2.7.16 ドキュメント (<https://docs.python.jp/2.7/>)
- ✓ Python 3.6.9 ドキュメント (<https://docs.python.jp/3.6/>)
Python のヘルプ ドキュメントです。
- ✓ Esri 製品サポート サイト (<https://www.esri.com/support/esri/>)
製品のダウンロード、FAQ やサンプル コードの参照、ドキュメントのダウンロードについてはこちらをご利用ください。※ログインが必要です。
- ✓ ArcGIS Desktop ヘルプ (<http://desktop.arcgis.com/ja/desktop/>)
ArcGIS 製品の使用方法や、提供される機能について知ることができます。また ArcPy を使用するのに必要となる、ジオプロセッシング ツールの構文が記述されています。
- ✓ ArcGIS Pro (<http://pro.arcgis.com/ja/pro-app/>)
ArcGIS Pro のダウンロードや使用方法、提供される機能について知ることができます。また ArcGIS Pro で ArcPy を使用するのに必要となる、ジオプロセッシング ツールの構文が記述されています。
- ✓ ArcGIS ブログ (<https://blog.esri.com/>)
ESRIジャパンの公式ブログです。
- ✓ GeoNet 開発ブログ (<https://community.esri.com/groups/devcom-jp>)
ArcGIS 開発者コミュニティで公開している開発ブログ記事です。
- ✓ ESRIジャパン GitHub (<https://github.com/EsriJapan>)
サンプルコードや関連資料などを公開しています。

サービス

- ✓ トレーニング サービス (<http://www.esri.com/training/>)
Esri 製品の技術を短期間で習得していただくためのトレーニング コースを実施しております。ArcPy を学習するためのコースも用意しております。
- ✓ 開発者サポート サービス (<http://www.esri.com/services/dev-support/>)
保守サービスに含まれる Q&A サポートの範囲を超えた、開発者向けの Q&A サービスを

有償にてご提供します。ArcPy に関する Q&A サポートはこちらをご利用ください。

✓ コンサルティング サービス (<http://www.esri.com/services/consulting/>)

GIS を導入するユーザー様、およびそれを支援するシステムインテグレータ様などの業務をプロフェッショナルが支援するサービスです。

- ・ 本書に記載されている内容は予告無く変更される場合があります。
- ・ 本書の一部または全部を無断で転用または複製することを禁じます。
- ・ 本書の内容に関しての電話でのお問い合わせは、お受けしておりません。
- ・ ArcGIS Desktop Basic、ArcGIS Desktop Standard、ArcGIS Desktop Advanced、ArcGIS、ArcMap、ArcCatalog、ArcToolbox、ArcGIS Pro、ArcGIS Enterprise、ArcGIS GIS Server、ArcGIS Engine、ArcGIS Online、Esri は、米国、欧州およびその他の管轄区におけるEsri 社の登録商標または商標です。
- ・ Adobe® Reader® は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。
- ・ Microsoft®, Office®, Access®, Excel® および Windows® は、米国Microsoft Corporation の、米国、日本およびその他の国における登録商標または商標です。
- ・ 本書に記載されている会社名、製品名は各所有者の登録商標および商標です。

書名	: ArcPy スタートアップガイド
発行日	: 2019 年 7 月 24 日 第 7 版
発行	: ESRIジャパン株式会社 〒102-0093 東京都千代田区平河町2-7-1 塩崎ビル
電話	: 03-3222-3941
FAX	: 03-3222-3946
URL	: http://www.esrij.com/