RPA学習コース

第２回目：UiPathでロボプロセスをサクッと作ってみる

2021年3月18日 19時

1HB-8S 黄　蔚菁

前回のおさらい
第 1 回目：RPA 紹介と無料な UiPath 開発環境構築

- RPA概要
  - RPAとは／RPAデモ／RPAが求められる理由／伝統的なプログラミング言語と比較
- 無料なUiPath開発環境構築の方法を把握した
- 録画機能で初めてのロボを作ってみた

前回の宿題

- まだStudioのインストールを完了していない方はスライド「10分で無料な開発環境構築してみよう」を参照して、次回の勉強会までに完了してください。（必須）

- 普段の生活や仕事の中で、ロボ化できそうなことってありますか。いくつか考えてみてください。（オプション）

本日の勉強会を終えると、あなたは…

- セレクターで画面UI要素の操作ができる。
- 変数を利用して、アプリケーションの間にデータ連携ができる。
- 分岐／繰り返し／サブフローを使って処理フローの制御ができる。
- 上記知識を用い、こんなロボを作れるようになる（デモ）

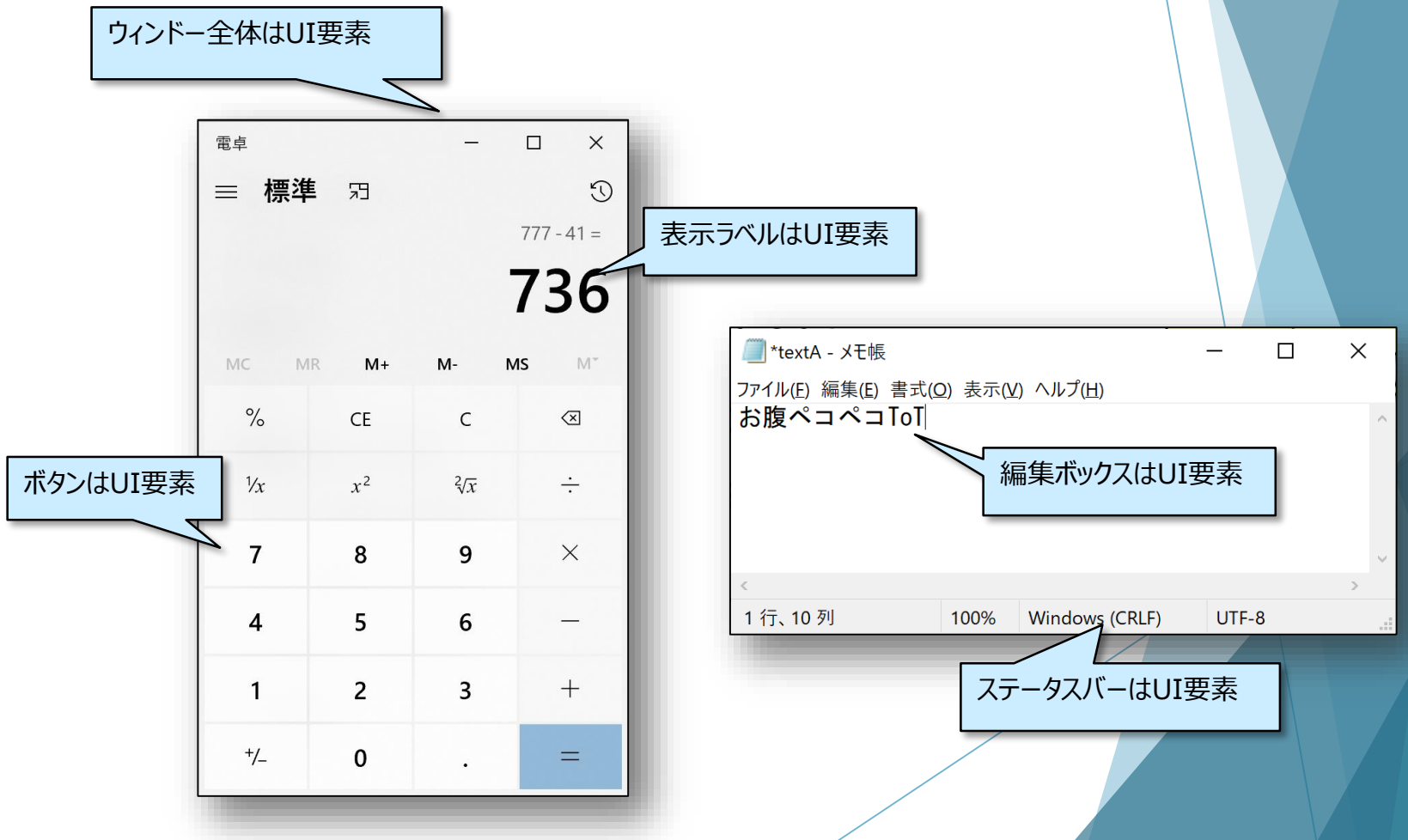
アジェンダ

- UI要素の操作
- アプリケーション間のデータ連携
- フロー制御
- 宿題

UI要素の操作

UI要素とは

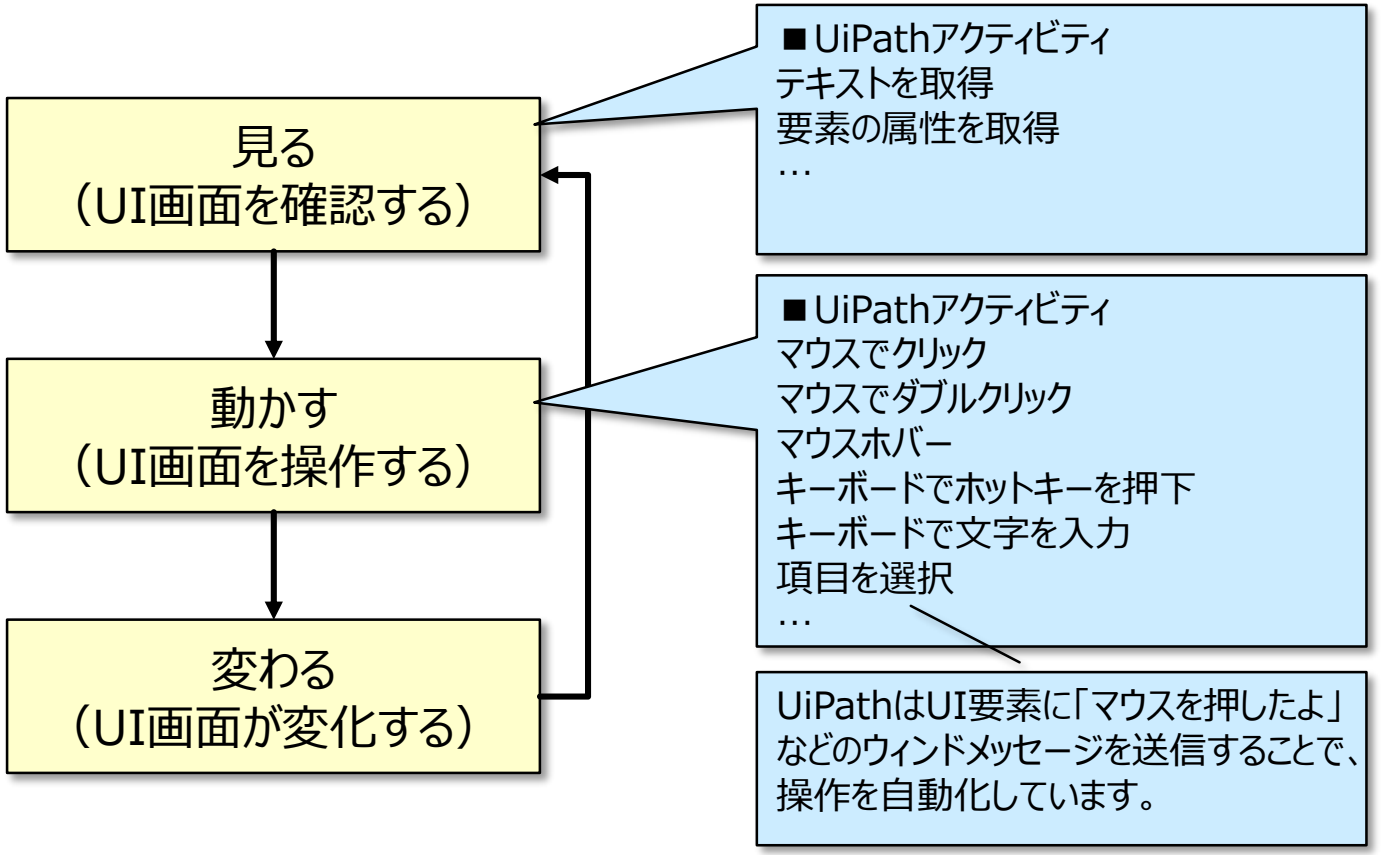
- ウィンドー
- ラベル
- ボタン
- メニュー
- 編集ボックス
- リストボックス
- コンボボックス
- スクロールバー
- その他

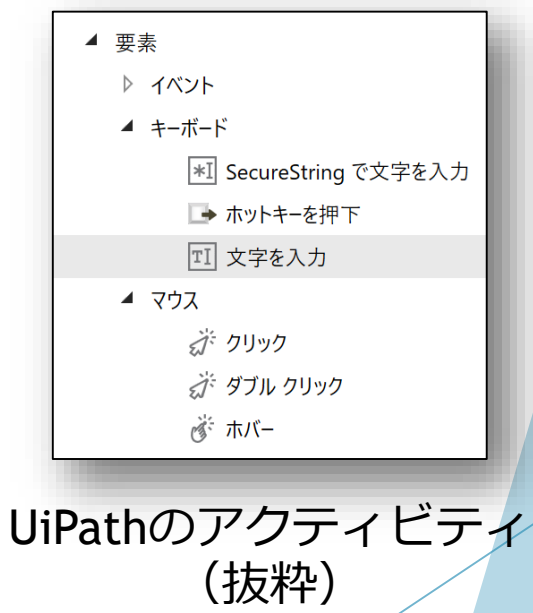


UI要素の操作
UiPathアクティビティ

■どんなUI操作でも、分解すると下記３種類の動作になる。

「見る」、「動かす」ためのUiPathアクティビティが大量に提供されている（皆さんの武器になる）

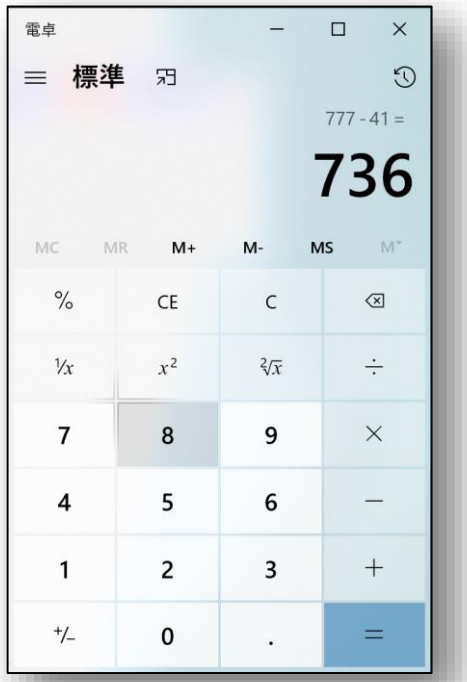




UiPathのアクティビティ
（抜粋）

UI要素の操作
セレクターとは

- セレクターは、アクティビティが操作対象とする**UI要素を特定するためのテキストデータ**。操作したいUI要素を選択（セレクト）するからセレクターと呼ばれる。




```
<wnd app='applicationframehost.exe' appid='Microsoft.WindowsCalculator_8wekyb3d8bbwe!App' title='電卓' />
<uia cls='LandmarkTarget' />
<uia automationid='NumberPad' cls='NamedContainerAutomationPeer' name='数字パッド' />
<uia automationid='num8Button' cls='Button' name='8' />
```

電卓アプリの「ボタン８」を表すセレクター

「ボタン８」のセレクター ＋ マウスクリック アクティビティ ⇒ 「ボタン８」をクリック

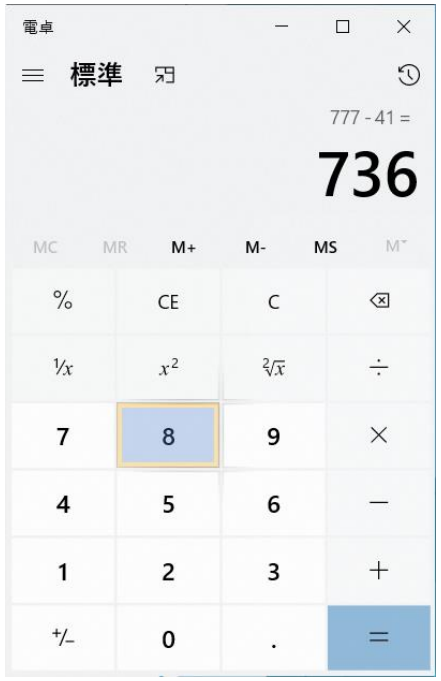
セレクタの使用例その一

UI要素の操作

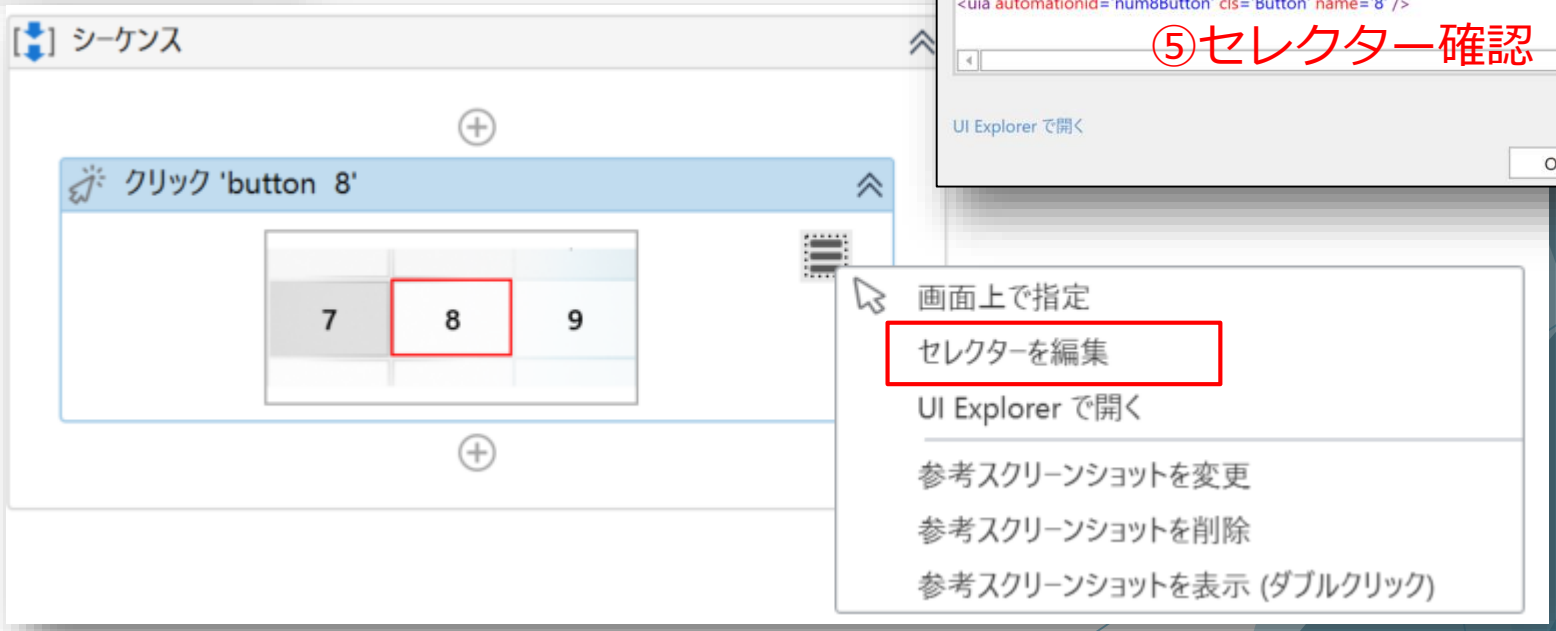
UI要素のセレクターを取得するのには

①ドラッグ＆ドロップでアクティビティ追加

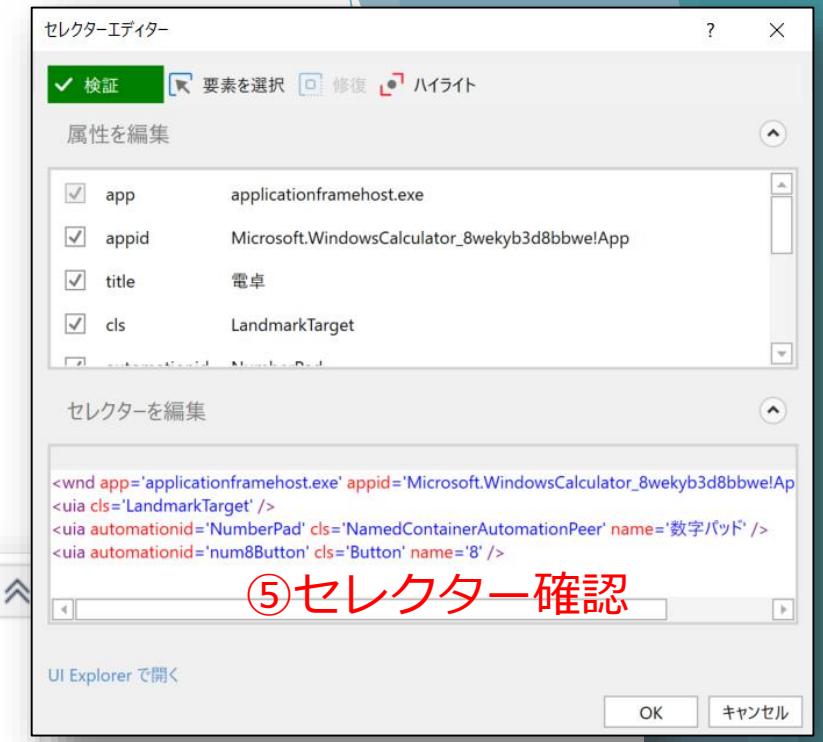
②「画面上で指定」をクリック



③操作したいUI要素をクリック



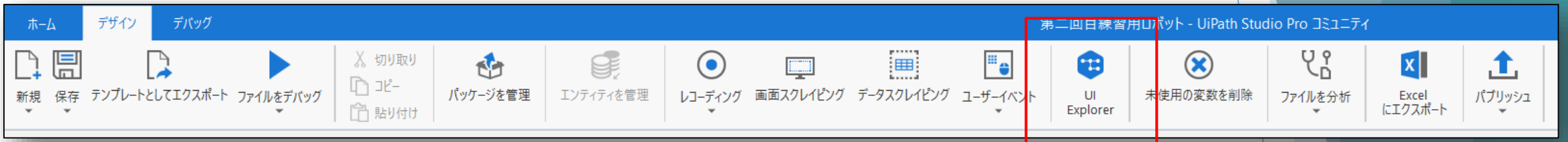
④セレクター編集で取得したセレクターを確認

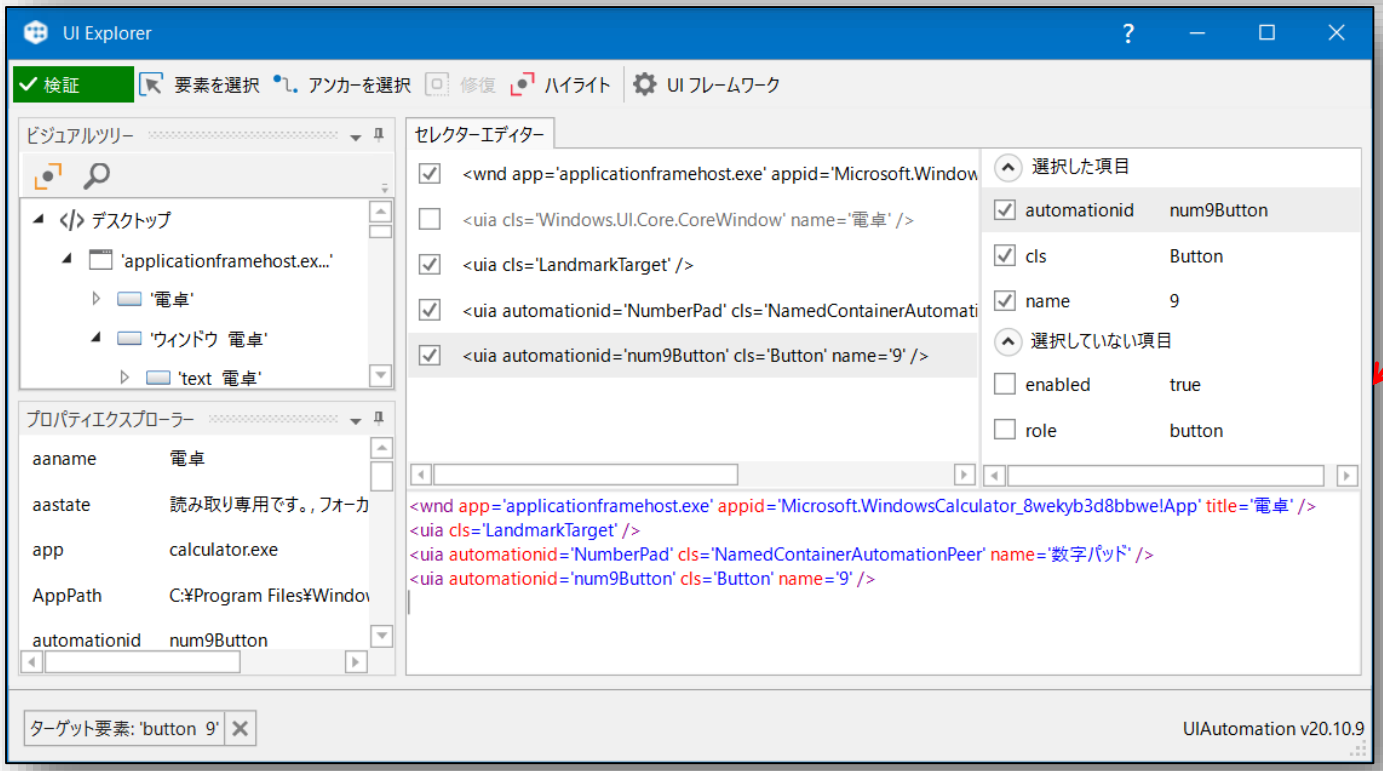


⑤セレクター確認

UI要素の操作

UI要素のセレクターを取得するのには





UI Explorer

UI要素の操作

自動取得したセレクターを編集する

```
<wnd app='notepad.exe' cls='Notepad' title='textA - メモ帳' />
<wnd aaname='水平' cls='Edit' />
<ctrl name='テキスト エディター' role='editable text' />
```

メモ帳を編集する処理がある。編集ボックスを特定するための、左記のセレクターは、textAというファイルが開かれた場合のみ有効になるため、宜しくない一例となる。

```
<wnd app='notepad.exe' cls='Notepad' />
<wnd aaname='水平' cls='Edit' />
<ctrl name='テキスト エディター' role='editable text' />
```

Titleを削除することで、titleにどんな文字が入っても、対象として識別される。

```
<wnd app='notepad.exe' cls='Notepad' title='*' />
<wnd aaname='水平' cls='Edit' />
<ctrl name='テキスト エディター' role='editable text' />
```

ワイルドカードを指定することで、titleにどんな文字が入っても、対象として識別される。

TIPS：セレクターの定義をよ～く観察して、開発環境に依存しない、かつ対象UI要素を十分に特定できるセレクターを定義することが大事！

UI要素の操作

練習時間

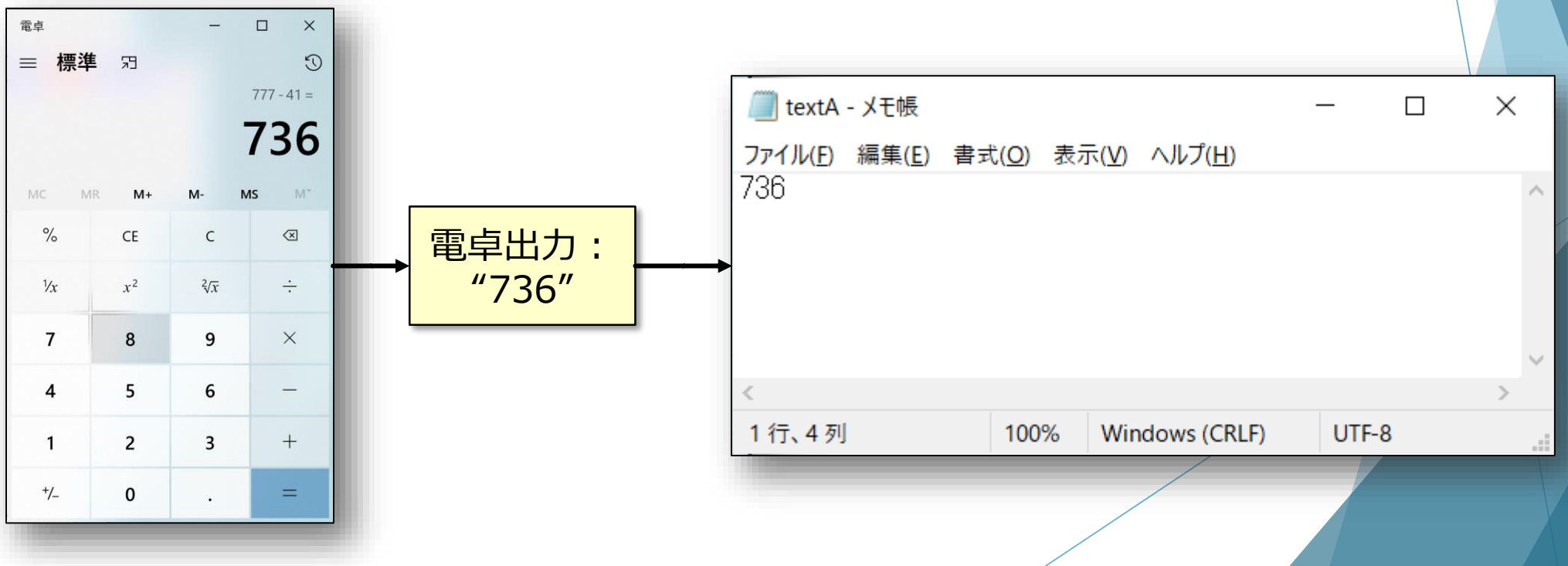
～～練習時間（5分間）～～

①マウスクリックアクティビティを利用して、電卓の「ボタン８」をクリックする。

②セレクターのテキストを直して、「ボタン9」を押すように修正してください。

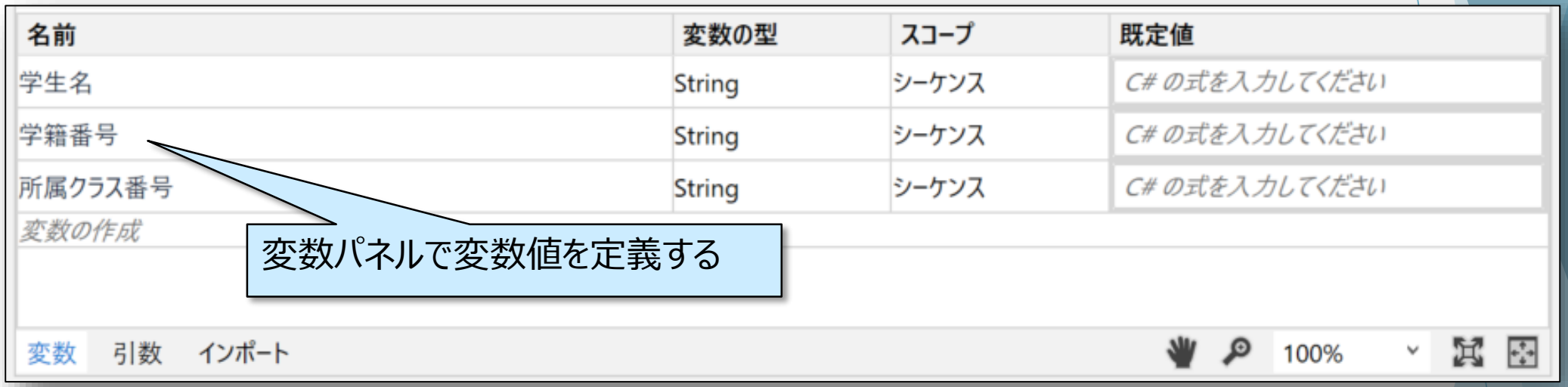
アプリケーション間のデータ連携
変数について

- 処理と処理の間で引き渡されるデータは、それぞれを区別できるように、名前がついた箱に入れられる。この箱を変数という。変数といっても、数値とは限らない。文字列や日付など、いろいろな種類のデータを変数で扱うことができる。



アプリケーション間のデータ連携

変数を定義する



TIPS：変数名は日本語で定義しても構わない！むしろ日本語を推奨する！

アプリケーション間のデータ連携
よく見かける変数タイプ

- 変数と引数の設定

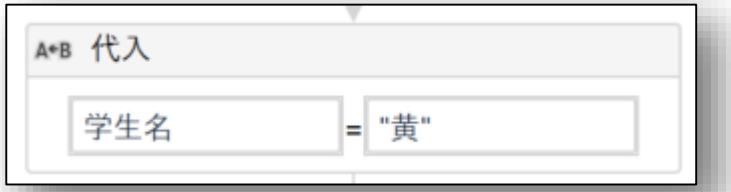
データタイプ	型	
整数	Int32	123
小数	Double	421.5
文字列	String	“がちゃがちゃ”
論理	Boolean	true/false
期間	TimeSpan	1.01:30:00.100（1日間1時間30分0秒と100ミリ秒）
日時	DateTime	2021/03/18
配列	Array of [T]	new String[] { "abc", "cde" }

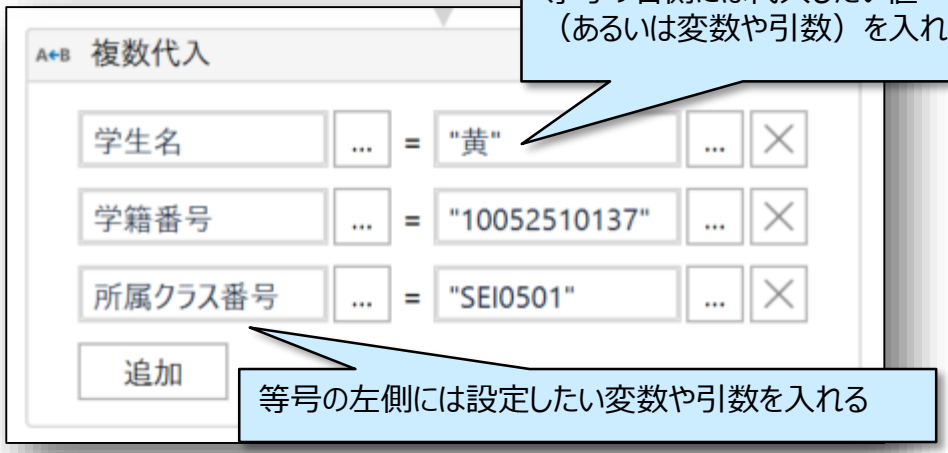
配列は頻繁に使われているが、コレクションと合わせて、次回に紹介させていただく。

★UiPathは.NET Framework との親和性が高く、NETの各種データタイプはそのまま使用可能
データタイプ辞書：Microsoft社の.Net Framework公式リファレンス
https://docs.microsoft.com/ja-jp/dotnet/api/system.string?view=net-5.0

アプリケーション間のデータ連携
変数の代入

- 代入
- 複数代入
- 情報取得用のアクティビティで設定

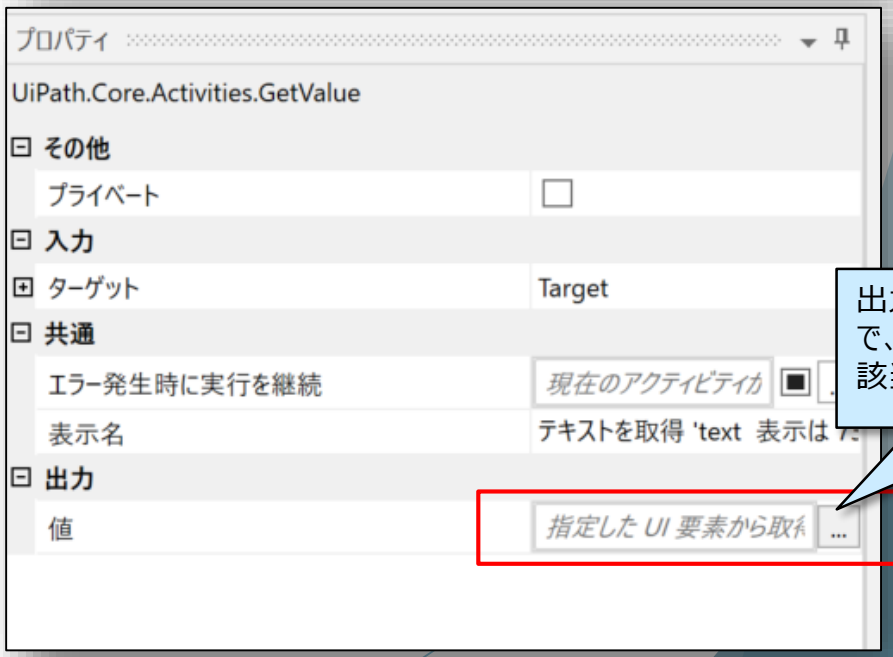




等号の右側には代入したい値
（あるいは変数や引数）を入れる

等号の左側には設定したい変数や引数を入れる

アクティビティ「テキストを取得」

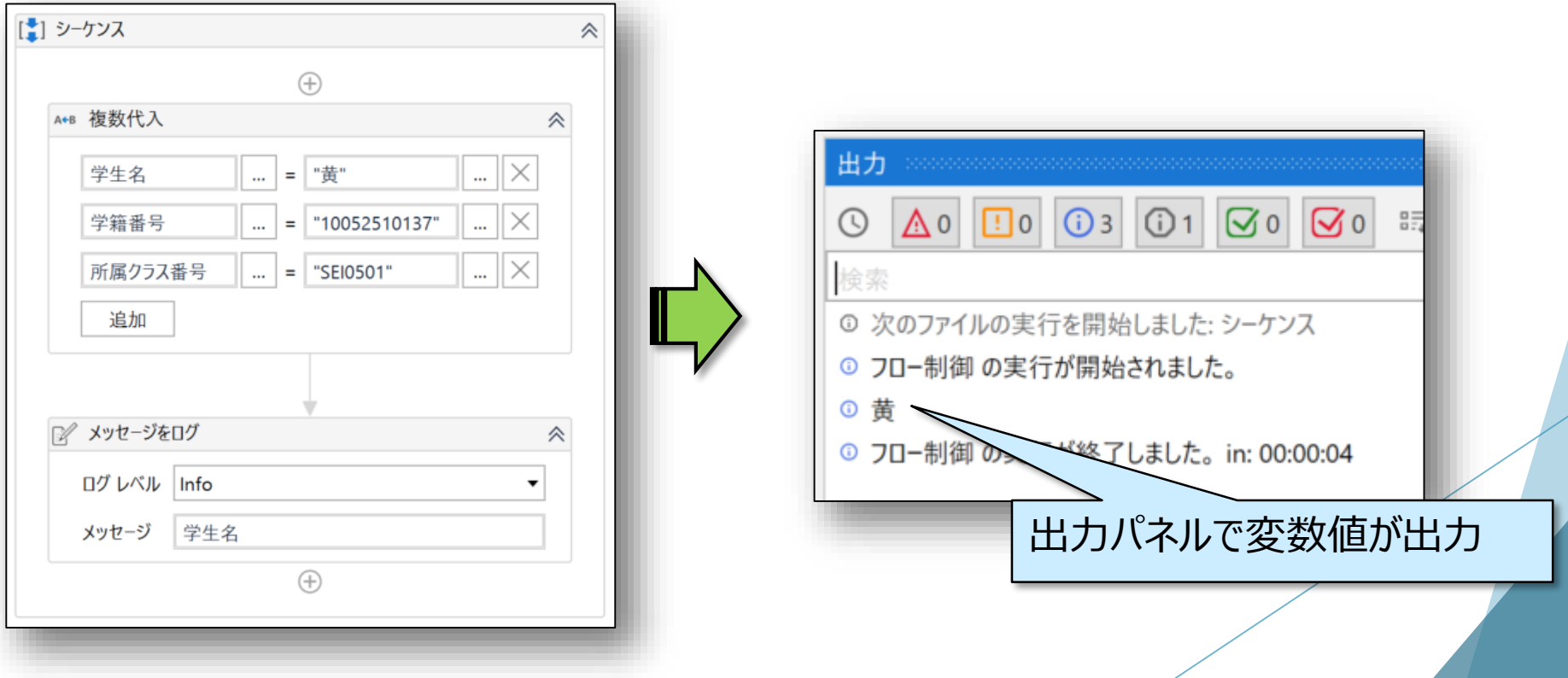


出力欄に変数や引数を入ることで、実行後に取得したテキストが該当引数や変数に代入される。

アプリケーション間のデータ連携
変数中身の確認

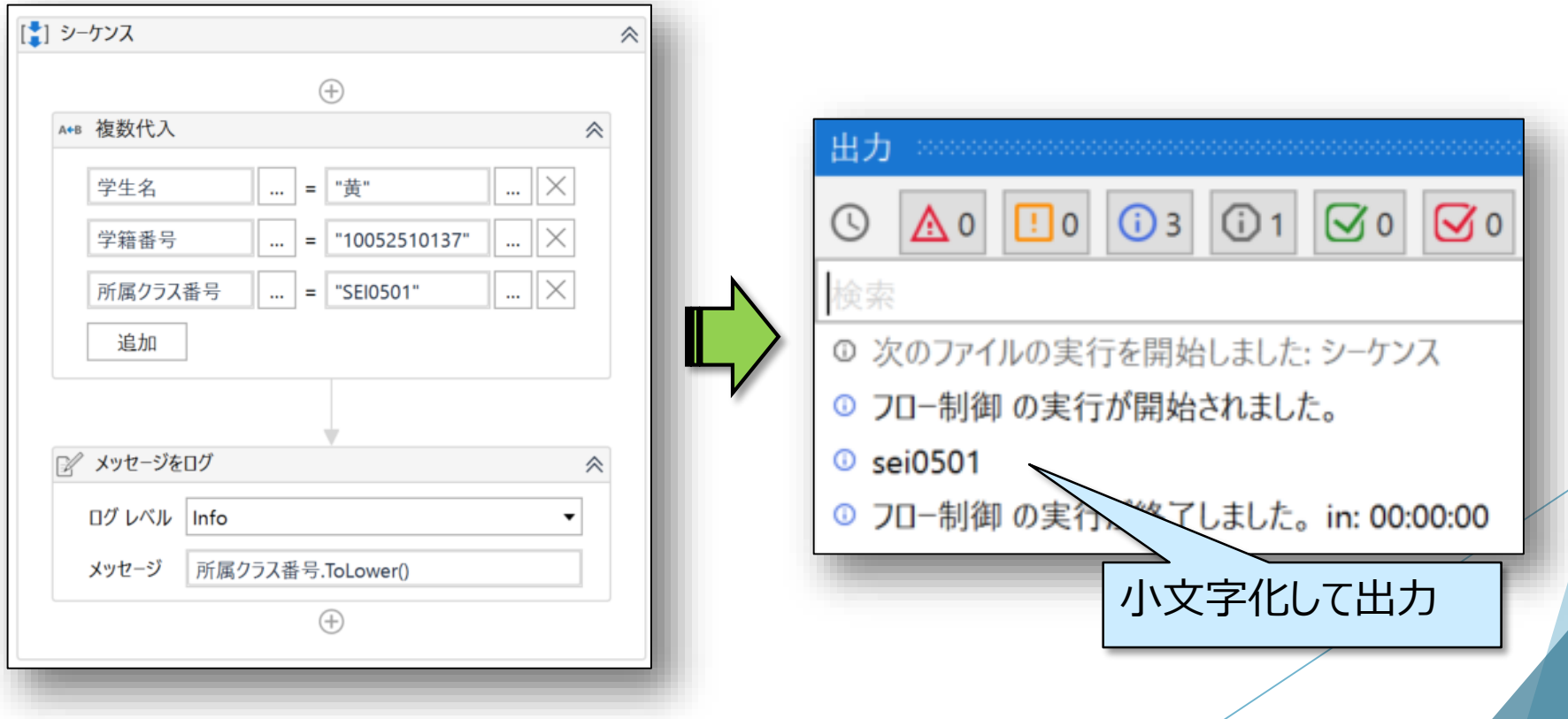
- 変数の中身を見てみたい時は？

→「メッセージをログ」アクティビティでプリントアウトしてみよう！



アプリケーション間のデータ連携
変数のメソッドを活用する（オプション）

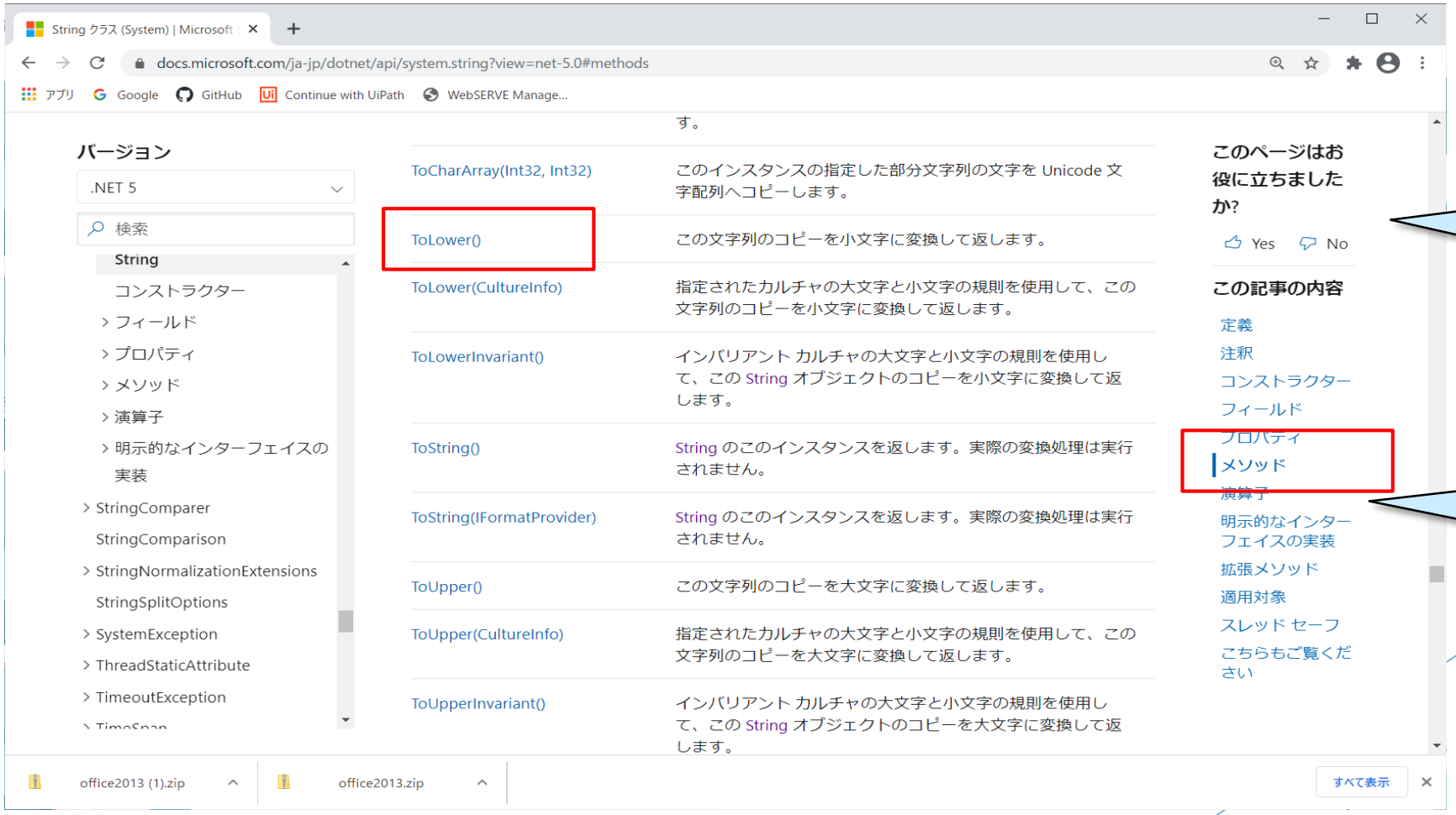
- 前述したようにUiPathは.Net Frameworkのデータタイプをそのまま使用しているので、.Netの提供メソッドはすべて使える。



アプリケーション間のデータ連携

変数のメソッドを活用する（オプション）

https://docs.microsoft.com/ja-jp/dotnet/api/system.string?view=net-5.0



Microsoft社の公式.Net Frameworkリファレンス

先ほどのString.toLower()の仕様が詳しく書かれている。

アプリケーション間のデータ連携
変数をセレクターに入れる

- 変数をセレクターに入れることで、ダイナミックにUI要素の特定ができる。

```
<wnd app='applicationframehost.exe' appid='Microsoft.WindowsCalculator_8wekyb3d8bbwe!App' title='電卓' />
<uia cls='LandmarkTarget' />
<uia automationid='NumberPad' cls='NamedContainerAutomationPeer' name='数字パッド' />
<uia automationid='num{{入力値}}Button' cls='Button' name='{{入力値}}' />
```

{{変数名}}でセレクターに変数を設定できる。
上記の定義で変数：「入力値」という変数の値が変わると、セレクターで特定する電卓ボタンも変わる

変数とセレクタの組み合わせ利用はUiPathのコアスキルである。やり方をぜひ覚えてください！

アプリケーション間のデータ連携
練習時間

～～練習時間（5分間）～～

①メモ帳の内容を取り出してログに出力してください。
②メモ帳に入れられた数字で電卓ボタンを押す処理を作ってみよう。

フロー制御

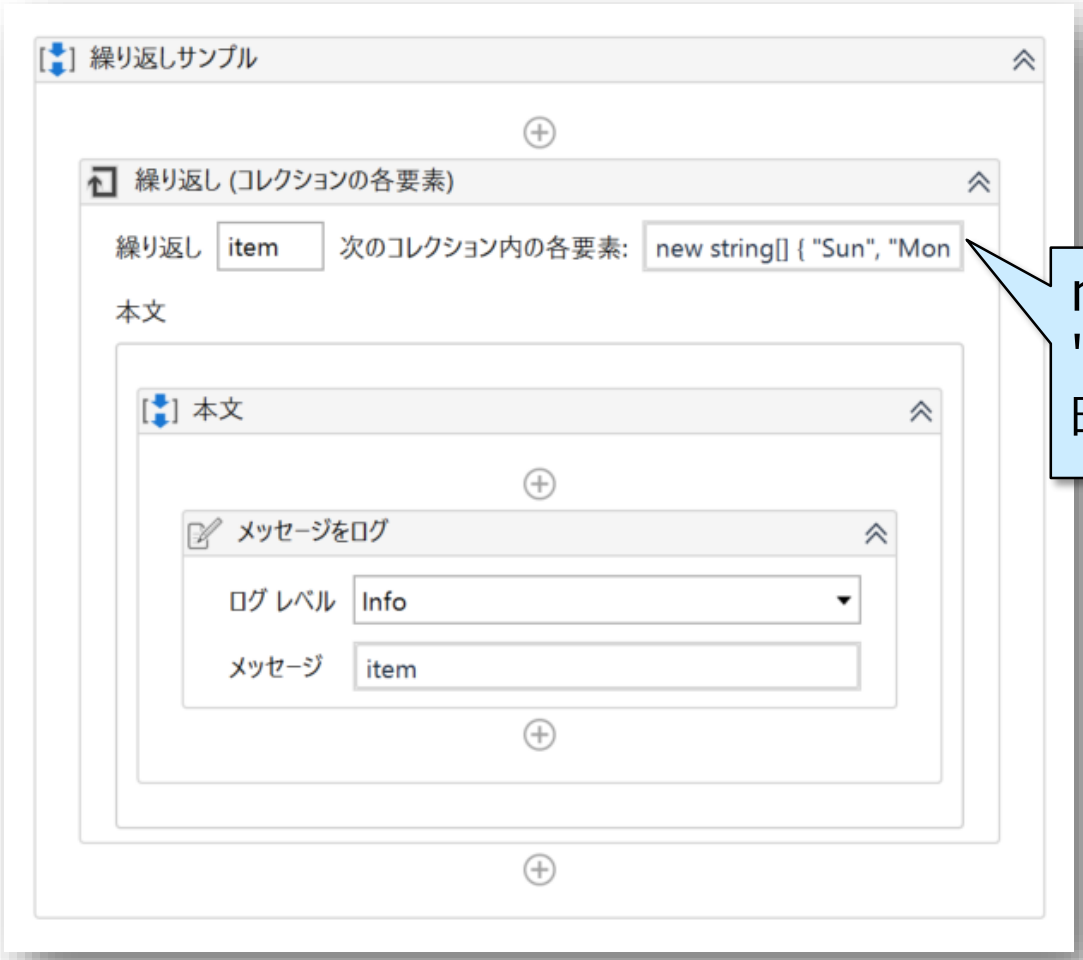
- 下記のように、プログラムと同じようなフロー制御をUiPathも提供している。これらを組み合わせることによって、どんなフローでも作れる。本日はデモをお見せしながらフロー制御を皆さんにマスターして頂く。

Java/C#の世界でこう呼ぶ	UiPathの世界でこぅよぶ
IF文	条件分岐
FORループ	繰り返し
メソッド	サブフロー

フロー制御
条件分岐のサンプル

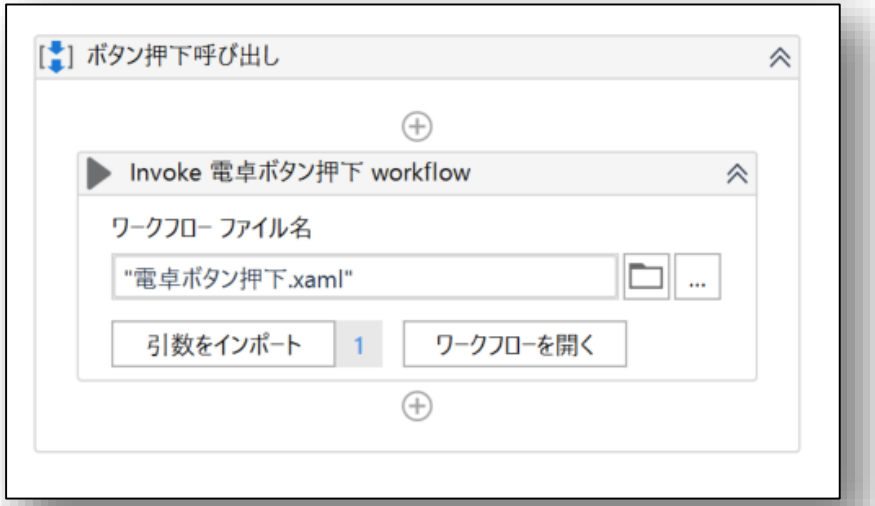
メモ帳の入力値を取得し、21点を超えたら、「惜しい、バストした！」越えてなければ、「まだセーフだ！」とポップアップされる。

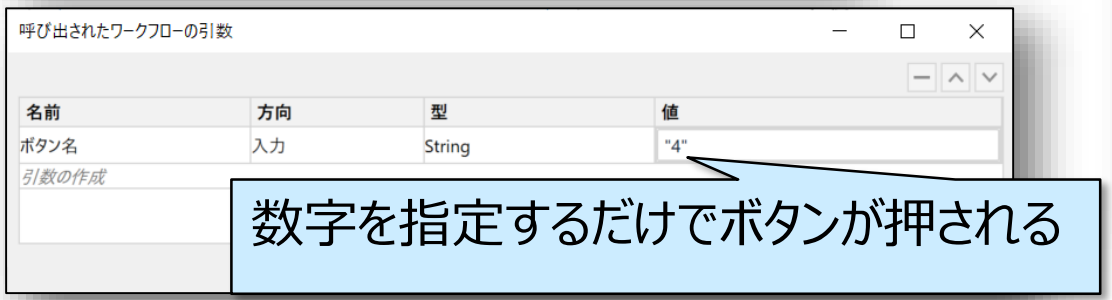
フロー制御
繰り返しのサンプル



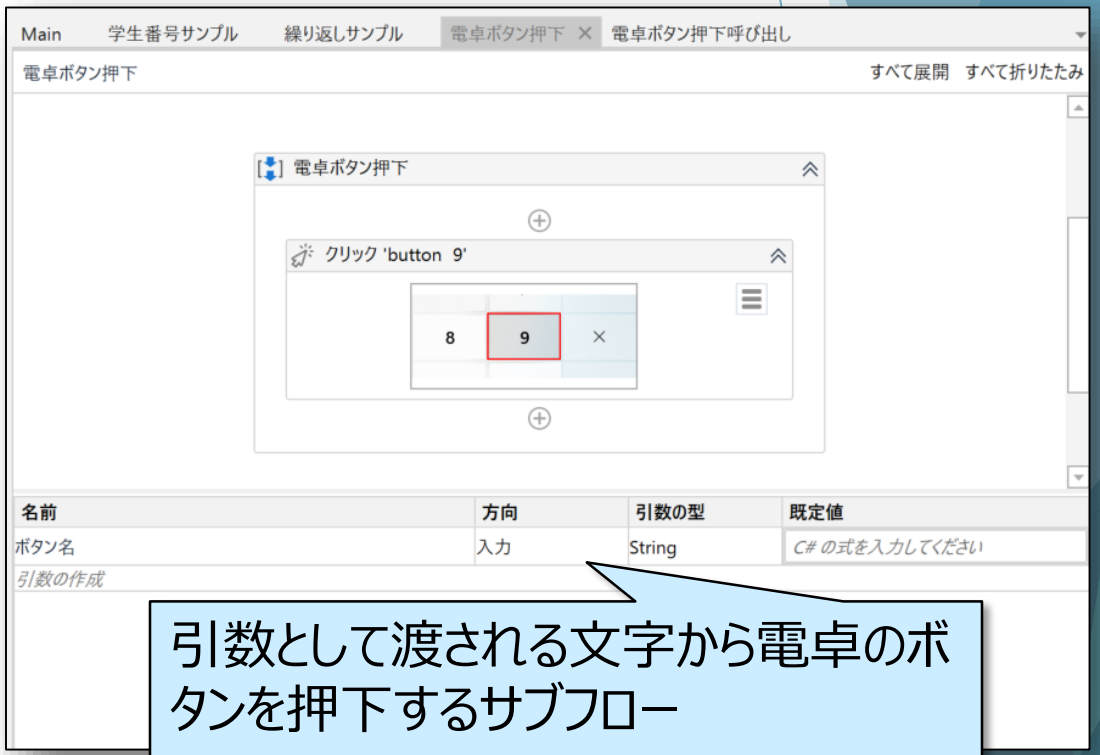
new string[] { "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" }
曜日の英語略をログに順番に出力するサンプル。

フロー制御
サブフローサンプル





数字を指定するだけでボタンが押される



引数として渡される文字から電卓のボタンを押下するサブフロー

フロー制御
サブフローを利用するメリット

- １つのワークフローで取り扱う問題を小さくする。
- 重複する処理を集約して、同一のワークフローとして部品化する。
- ワークフロー単位でテストができるようになる。
- チーム開発を容易にする。

フロー制御

練習時間

～～練習時間（10分間）～～

メモ帳に書かれている計算式を拾い、電卓に移して計算し、
結果をまたメモ帳に転記するロボットを作ってください。

フロー制御

練習時間

～～練習時間（10分間）～～

ヒント１：電卓の「＋－×÷」ボタンのセレクターを確認してください。

ヒント２：Stringを繰り返しアクティビティに入れたら、文字毎に繰り返される。

質問コーナー

宿題

- スライド「フロー制御：練習時間」の電卓転機ロボを完成してください。

次回予告

第三回目：高度なUiPath機能（１）（4/1（木） 19時）

- データテーブルとコレクション
- エラーハンドリング
- 並列とキャンセルスコープ
- C#でカスタマイズ処理を作ってみよう

次回からは皆さんの武器を増やす会です！

では、ご参加お待ちしております！！