

『作ろう！ハンズオントレーニング』



NTTアドバンステクノロジー株式会社

はじめに



本資料の構成と学習目標

- 本資料の構成および学習目標は以下のとおりです。



目次

学習目標

第1章 WinActor の基本操作

WinActorの基本操作

WinActorの基本操作を理解する



- 本資料の構成および学習目標は以下のとおりです。

第2章 シナリオ作成		目次	学習目標
基礎編	Webページのクリック&入力を自動化する	Webページでの利用方法を理解する 画像マッチングの利用方法を理解する	
	WebページからExcel台帳への転記を自動化する	Webページでの利用方法を理解する Excelでの利用方法を理解する	
	Excel台帳から社内システムへの転記を自動化する	社内システムでの利用方法を理解する Excelでの利用方法を理解する	
	Excel台帳から別のExcel台帳への転記を自動化する	Excelでの利用方法を理解する	
	メール送信を自動化する	メールの送信方法を理解する	

- 本資料の構成および学習目標は以下のとおりです。

第2章 シナリオ作成		目次	学習目標
実践編	何度も同じ処理を繰り返し行う		繰り返し処理の利用方法を理解する
	取得した文字や数字を加工する		加工方法の種類を理解する
	条件によって処理内容を変更する		条件実行処理の利用方法を理解する

- 本資料の構成および学習目標は以下のとおりです。

第2章 シナリオ作成		目次	学習目標
応用編	自動記録機能によるシナリオ作成について学ぶ	自動記録機能を使って、シナリオのベースを作成する	
	修正しやすいシナリオ作成方法を学ぶ	保守性の高いシナリオの作成方法を学ぶ	
	トラブル時の対応方法を学ぶ	トラブルが起きた際に、確認する事項や具体的な対応方法について学ぶ	

本書において以下に記載された名称、およびその他記載されている会社名、製品名は、各社の登録商標または商標です。なお、本文中ではTM、®、©マークは省略しています。

- WinActorはNTTアドバンステクノロジー株式会社の登録商標です。
- Microsoft、Windows※1、Microsoft Edge、Excel、VBScript※2は、米国Microsoft Corporation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。
 - ※1 Windowsの正式名称は、Microsoft Windows Operating Systemです。
 - ※2 VBScriptの正式名称は、Microsoft Visual Basic Scripting Editionです。
- その他の記載されている会社名、製品名は各社の商標または登録商標です。

- 本書および提供するソフトウェア類に付された著作権表示「Copyright © 2020-2026 NTT ADVANCED TECHNOLOGY CORPORATION」の変更、削除をすることはできません。
本書の著作権はNTTアドバンステクノロジー株式会社に帰属します。
- 本書では、Windowsの操作方法や機能を理解されていることを前提として説明しています。
本書に記載されていないことについては、Microsoftが提供しているドキュメントなどをご覧ください。

WinActor使用上の注意点およびインストール

- 本資料の学習でWinActorを使用する場合の注意点については、「WinActor 操作マニュアル WinActor の使用に関する注意」をご参照ください。
- 本資料の学習でWinActorを使用する場合のインストールおよびライセンスの登録方法については、「WinActor インストールマニュアル」および、「WinActor 操作マニュアル ご利用開始までの流れ」をご参照ください。
- ライセンスの種別については、「WinActor 操作マニュアル ライセンス種別と利用できる機能」をご参照ください。
- 本資料は、WinActor Ver. 7.5.0 以降を対象としています。

2

WinActor使用上の注意点およびインストール

- 本資料で学習を行う前に以下の手順を実施し、マニュアルのダウンロードを可能にした上で
 - 「シナリオの動きを見てみよう！動画」
 - 「ハンズオントレーニングで使用する各種ファイル」をダウンロードしてください。
マニュアル同梱インストーラーでインストールしたユーザーは以下の手順は不要です。

準備

マニュアルのダウンロード



1

WinActorを起動し、メニューバーの［ツール］ - ［オプション］からオプション画面を開きます。

2

WinActor使用上の注意点およびインストール

準備

マニュアルのダウンロード

2 「その他」タブをクリックします。

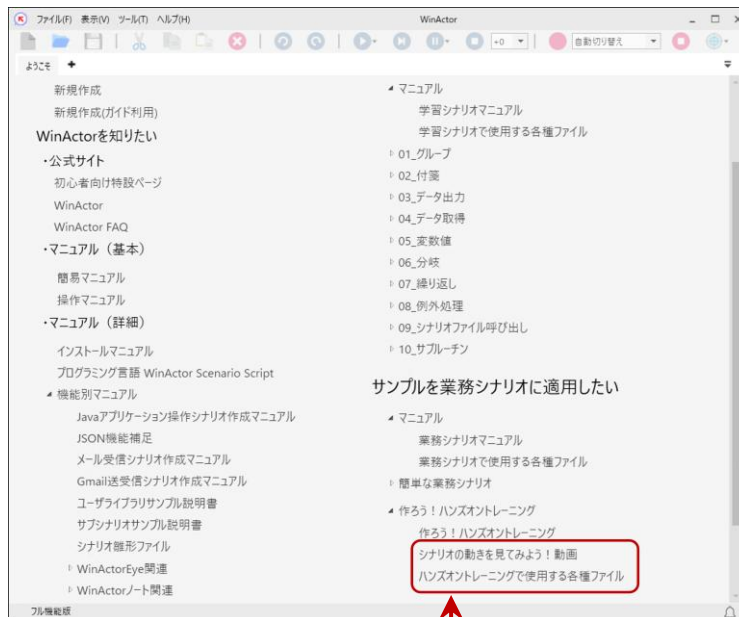
The screenshot shows the 'オプション' (Options) dialog box. At the top, there is a tab bar with the following tabs: '管理サーバ', 'プロキシサーバ', 'サーバ接続状況', 'ログサーバ', '実行', '記録', and 'その他'. The 'その他' (Others) tab is selected and highlighted with a red box. An arrow points from the instruction box to this tab. Below the tabs, there are several settings: 'ノード表示幅' (Node display width) set to 15, '表示ログ行数上限' (Maximum number of log lines to display) set to 1,000, 'フォントサイズ' (Font size) set to 12, a checked checkbox for 'CSVファイル読み込み時に連続する改行文字をまとめる' (Combine consecutive line breaks when loading CSV files), a checked checkbox for '最小化時にタスクバーにアイコンを表示する' (Display icon in taskbar when minimized), a dropdown menu for '古い作業ファイルの削除' (Delete old work files) set to '確認後に削除する' (Delete after confirmation), and a '削除しきい値 (日)' (Delete threshold (days)) set to 7. Below these settings is a note: '※WinActor起動時に、削除しきい値に指定した日数以上経過した古い作業ファイルが存在する場合、設定したアクションを実行します。ユーザ自身でWinActor作業フォルダにファイルを配置したことがなければ削除して問題ありません。' (When WinActor starts, if old work files that have exceeded the specified number of days exist, the set action will be executed. If you have not configured files in the WinActor work folder yourself, deleting them is not a problem). At the bottom, there is a checked checkbox for 'マニュアルのオンラインダウンロードを許可する (要インターネット接続)' (Allow online download of manual (requires internet connection)). An arrow points from the instruction box to this checkbox. At the very bottom, there are 'OK' and 'キャンセル' (Cancel) buttons. The 'OK' button is highlighted with a red box, and an arrow points from the instruction box to it.

3 「マニュアルオンラインダウンロードを許可する (要インターネット接続)」にチェックを入れ、

4 「OK」ボタンをクリックしてオプション画面を閉じます。

準備

マニュアルのダウンロード



5

上記いずれかのように画面が表示されるので、
「シナリオの動きを見てみよう！動画」
「ハンズオントレーニングで使用する各種ファイル」
をクリックします。

第1章 WinActorの基本操作



1

WinActor操作の概要

- WinActorでは、記録、編集、動作チェックを繰り返しながら「シナリオ」を作成します。作成した「シナリオ」の実行時には、必要に応じて投入データも読み込みます。



✓ シナリオを作成するには、フローチャートで操作を定義する必要があります。

✓ フローチャートを使って、“手動による操作の定義”もしくは、“自動で操作を記録”して、シナリオを作成することができます。

2

WinActorの基本操作

2-1 WinActorの起動

1

デスクトップ上の<起動アイコン>をダブルクリックします。



起動アイコンをクリックしても画面が表示されない場合、次の点をご確認ください。

1. WinActorは起動するまで“少し時間がかかります。”起動するまでお待ちください。
2. WinActorは二重起動できません。既に起動しているWinActorが最小化されて、タスクトレイに格納されている可能性がありますので、タスクトレイをご確認ください。

2-2 WinActorの画面構成

メニューバー

パレットエリア

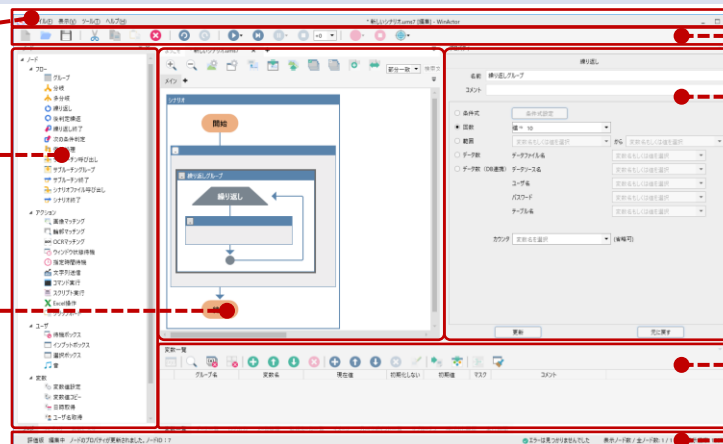
シナリオ編集エリア

ツールバー

プロパティエリア

機能編集エリア

ステータスバー



各画面の詳細は、操作マニュアル「3 WinActor の画面」をご参照ください。

2

WinActorの基本操作

2-3

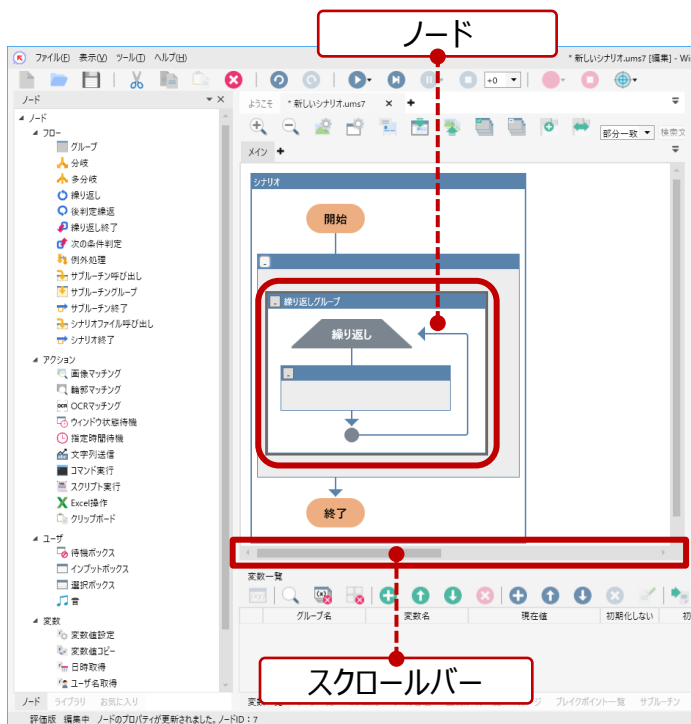
WinActorの画面要素



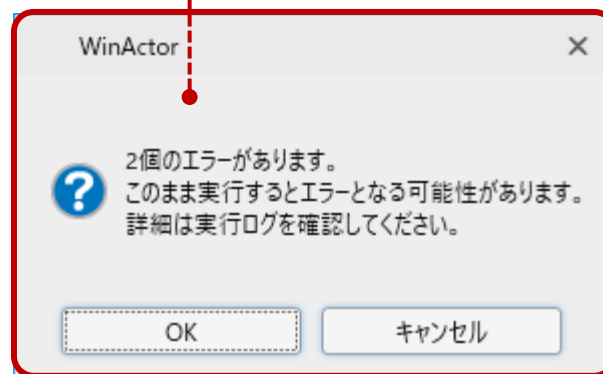
「ノード」とは、シナリオを構成する部品です。
フローチャート上は箱状で表示されます。



「ダイアログ」とは、情報表示、警告表示、情報入力
など、ユーザとの対話用に表示される画面です。



ダイアログ



スクロールバー



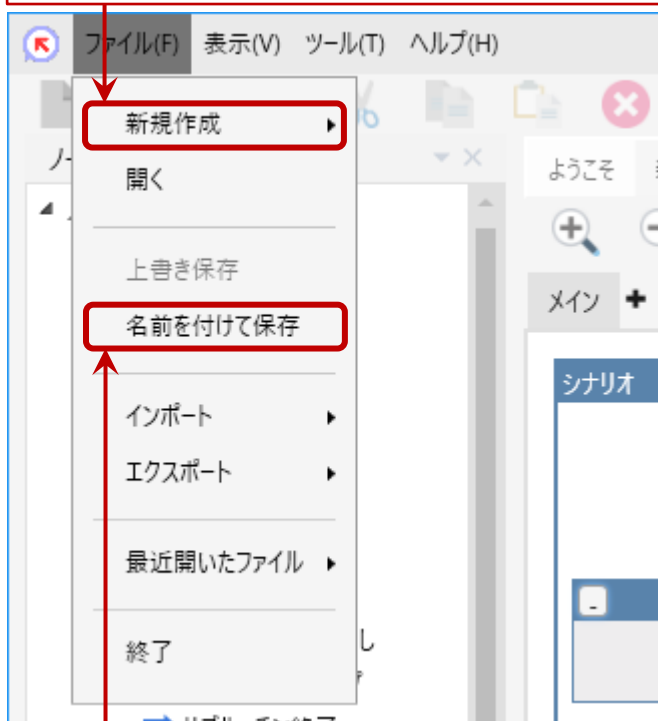
WinActorの画面要素の詳細は、操作マニュアル「7.2 画面要素と呼び名について」をご参照ください。

2

WinActorの基本操作

2-4 シナリオファイルの新規作成

1 メニューバーの[ファイル]-[新規作成]を選択します。



2 シナリオ作成後に[ファイル]-[名前を付けて保存]を選択します。

3 シナリオファイルを保存するフォルダを選択し、



4 <ファイル名> テキストボックスにシナリオファイル名を入力して、

5 <保存> ボタンをクリックします。

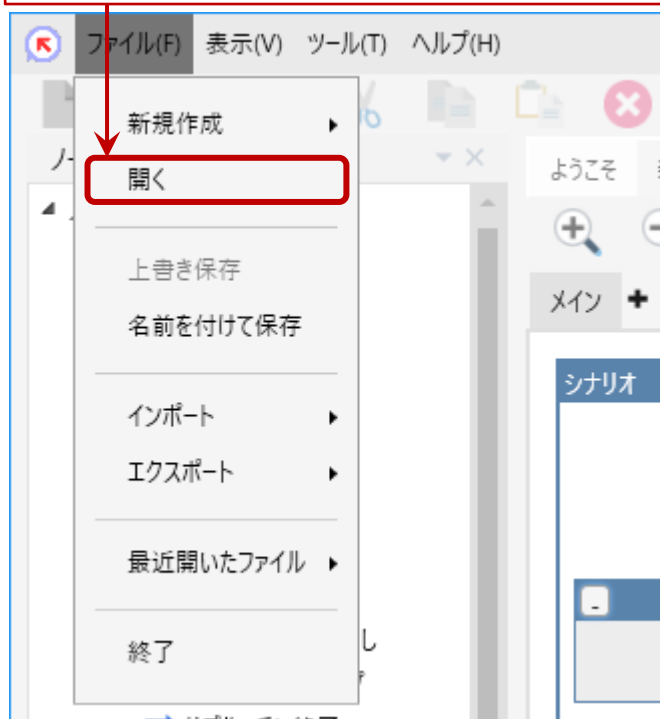
「シナリオ」は、Microsoft OfficeのWordやExcelと同じように、ファイル形式で保存できます。

2

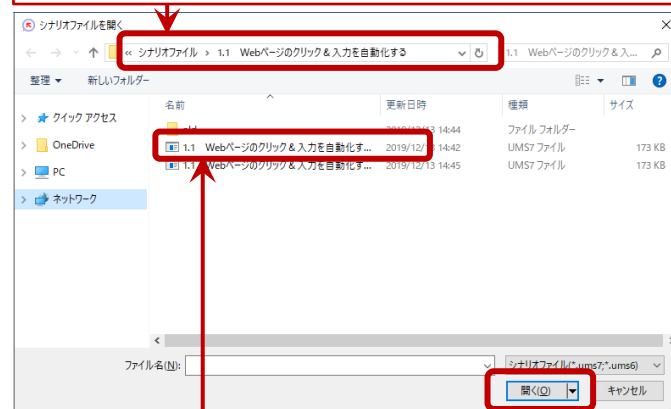
WinActorの基本操作

2-5 シナリオファイルの読み込み

1 メニューバーの[ファイル]-[開く]を選択します。



2 シナリオファイルが保存されているフォルダを選択し、



3 該当するシナリオファイルを選択して、

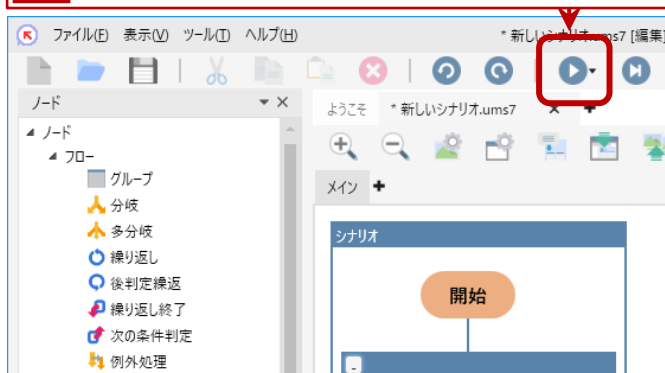
4 <開く> ボタンをクリックします。

2

WinActorの基本操作

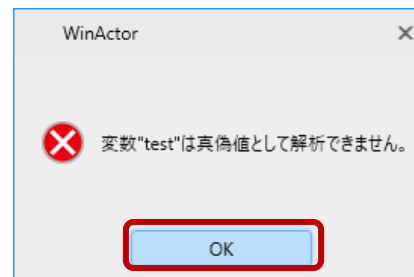
2-6-1 シナリオの実行

- 1 <シナリオ実行> ボタンをクリックします。

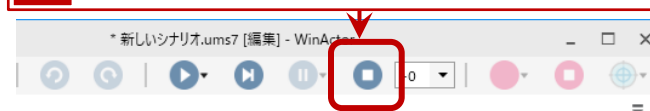


2-6-2 シナリオの実行時エラー対応

- 1 エラーが発生したら、ダイアログに表示されたエラーメッセージを確認し<OK> ボタンをクリックします。



- 2 <停止> ボタンをクリックします。



- 3 ログの確認や、プロパティ等の設定を見直します。



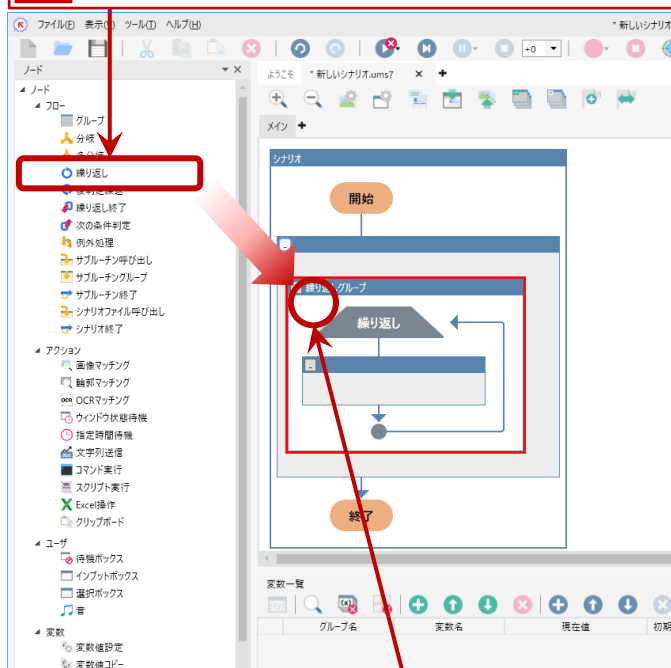
「ログ出力画面」の表示は、本章2-10を参考に確認してください。

2

WinActorの基本操作

2-7-1 手動によるシナリオ作成

1 ノードパレットから対象となる<ノード>をドラッグし、



2 シナリオ編集エリア上でドロップします。未入力/設定エラーがある場合、赤枠で表示されます。

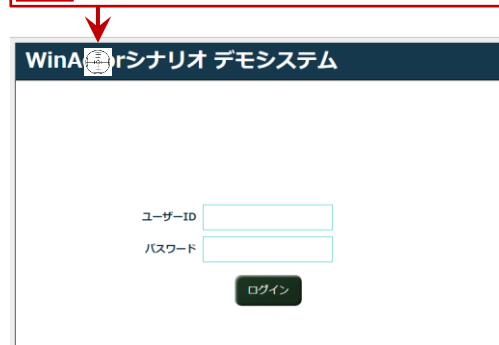
ノードの詳細は、操作マニュアル「4 ノード」をご参照ください。

2-7-2 自動（操作の記録）によるシナリオ作成

1 ツールバーの<記録対象アプリケーション選択>ボタンをクリックし、



2 自動（操作の記録）の対象をクリックします。

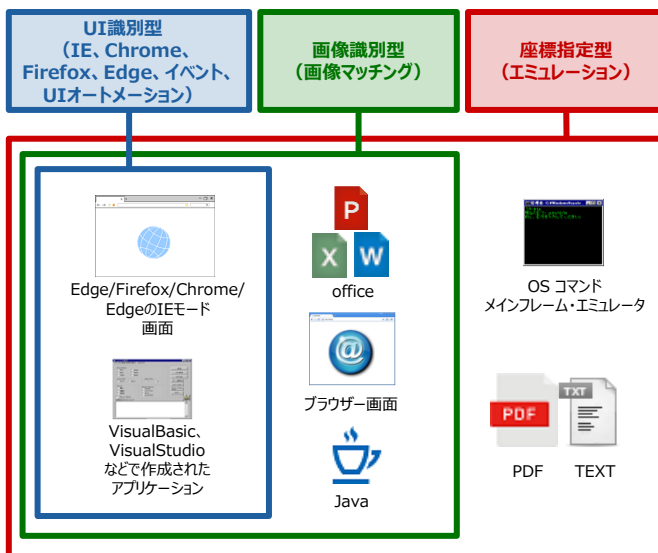


2

WinActorの基本操作

2-7-2 自動（操作の記録）によるシナリオ作成

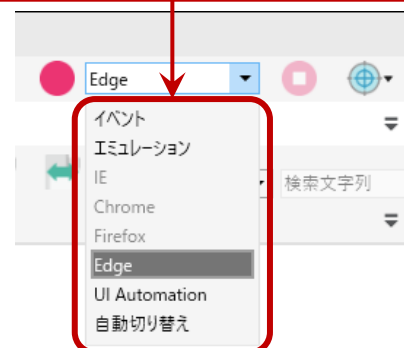
3 自動（操作の記録）の対象に応じて、以下の記録モードが自動選択されます。



4 選択されている記録モードを確認できます。



5 必要に応じて、＜記録モード＞の右の＜▼＞ボタンをクリックすると、使用する[記録モード]を手動で選択することもできます。



6 ＜記録＞ボタンをクリックします。



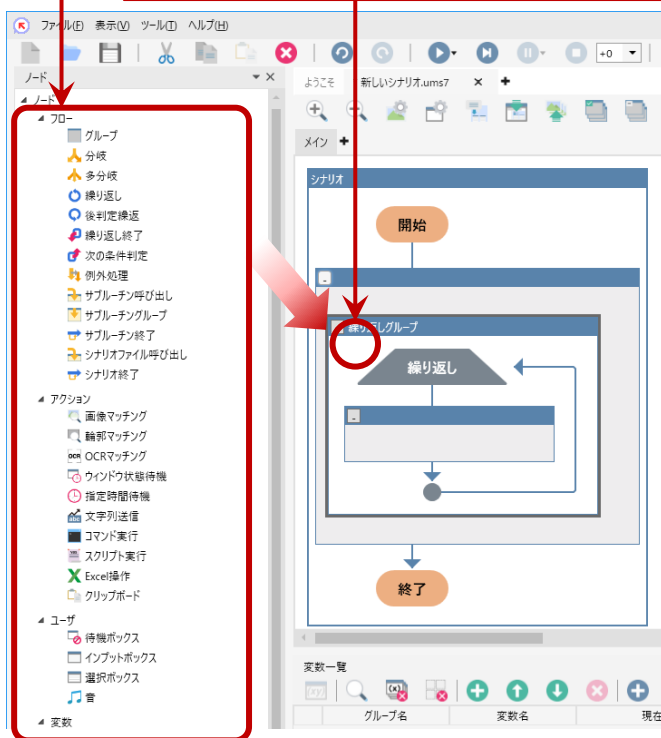
2

WinActorの基本操作

2-8-1 シナリオの編集

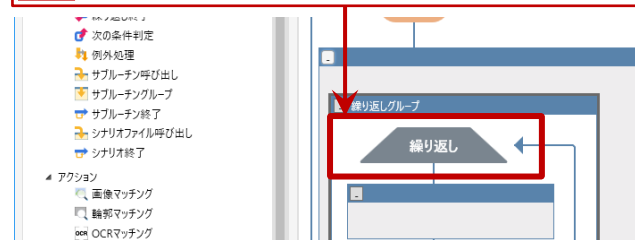
1 パレットエリアのノードパレットから追加する
ノードをドラッグし、

2 シナリオ編集エリア上の該当位置でド
ロップします。

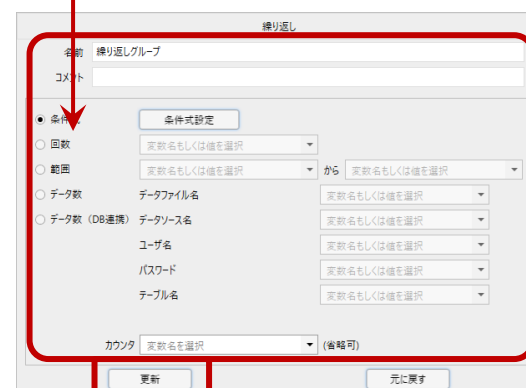


2-8-2 シナリオの編集（プロパティ画面）

1 シナリオ編集エリア上のノードをダブルクリッ
クし、



2 ノード種別ごとの設定をして、



3 <更新> ボタンを
クリックします。

2

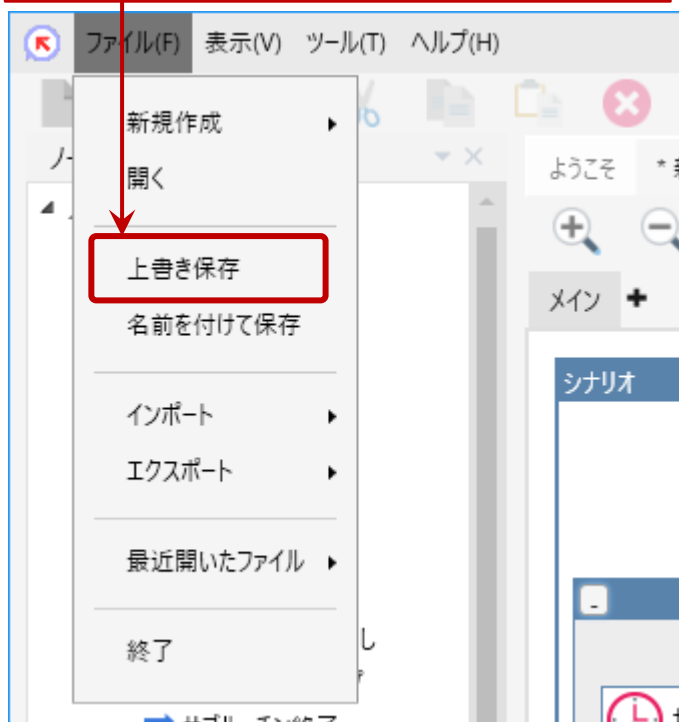
WinActorの基本操作

2-9

シナリオファイルの保存

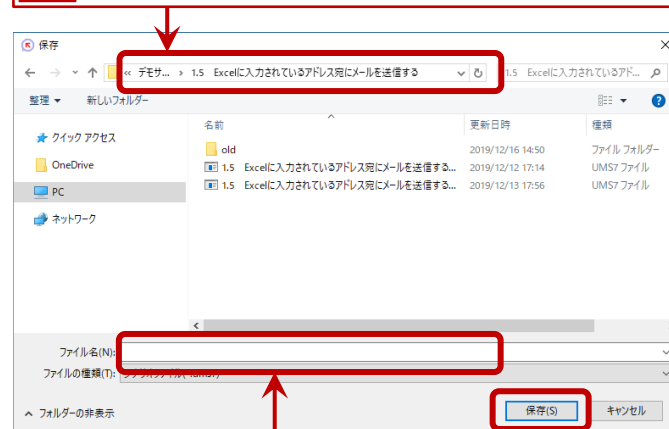
1

メニューバーの[ファイル]-[名前を付けて保存]または、[上書き保存]を選択します。



2

シナリオファイルを保存するフォルダを選択し、



3

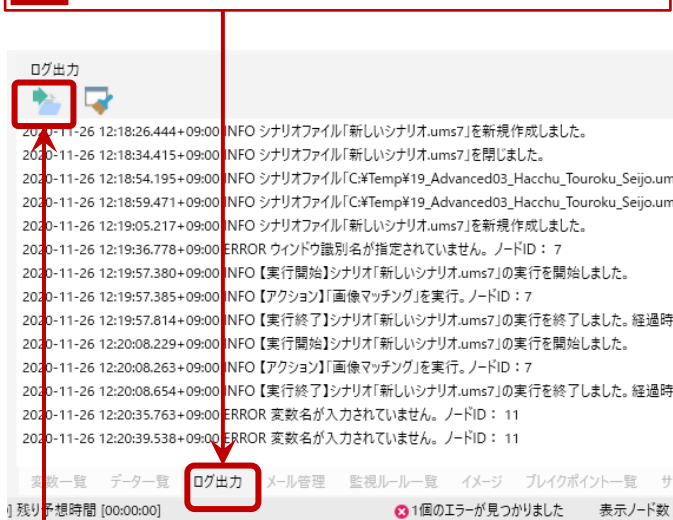
<ファイル名> テキストボックスにシナリオファイル名を入力して、

4

<保存> ボタンをクリックします。

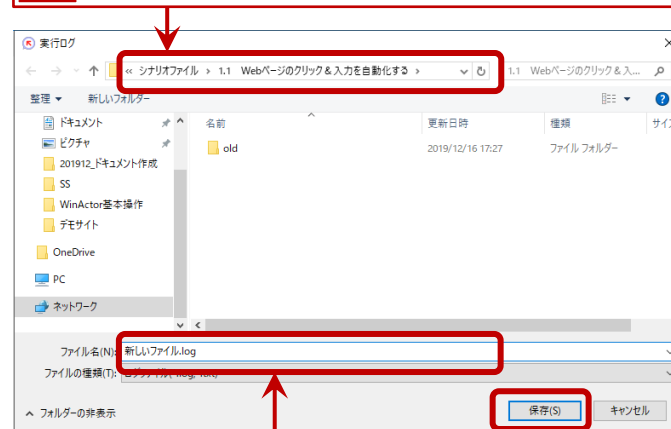
2-10 シナリオの実行ログ保存

1 機能編集エリアの[ログ出力]タブをクリックし、



2 <ログ保存> ボタンをクリックします。

3 ログファイルを保存するフォルダを選択し、



4 <ファイル名> テキストボックスにログファイル名を入力して、

5 <保存> ボタンをクリックします。



「ログ出力画面」の表示は、必須ではありませんが、エラーが発生した際のログも表示されるので、本資料でシナリオ実行を確認する際には、表示しておくことをお勧めします。

第2章 シナリオ作成



～ 基礎編 ～



1

シナリオ作成 ～基礎編プロローグ～

導入

これはとある会社の調達部の話です。

宇院(ういん)さんは調達部の2年目社員として、今まで行っていた定型作業の他に大きな仕事を任されるようになりました。しかし業務が増えたことにより作業ミスやモレが発生してきています。

この現状をどうにかしようと思い、WinActorを使って作業を自動化している増田(ますた)先輩に相談しました。みなさんも宇院さんと一緒にWinActorを使った自動化の一步を踏み出しましょう!!



宇院(ういん)さん

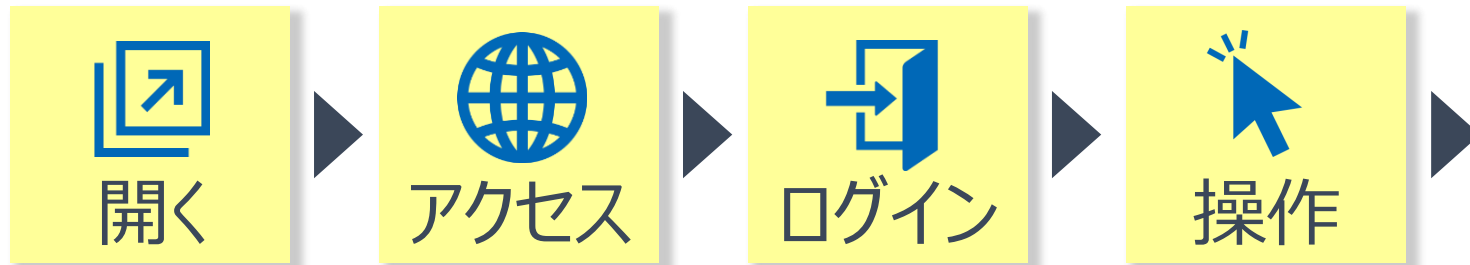
調達部で働く2年目社員。
だんだんと大きな仕事を任されるようになり、やる気が出てきている。



増田(ますた)先輩

既にWinActorを使用している他部署の先輩社員。
今回宇院さんの相談を受け、サポート役をかって出た。

基礎編



Webページのクリック&入力を自動化する

導入

宇院さんは発注管理台帳に記載されている品物を、毎週発注先の発注Webシステムに入力し発注する作業をしています。

最近は任される仕事も増え、忙しさもあり発注作業にミスが発生してきました…。



「毎週発注管理台帳に載っている未発注の品物を、発注先のWebシステムに登録するのが手間だなあ…。品数も多いし発注数を間違えると大変なことになってしまいます…。どうにかならないでしょうか？」



「業務システムなどへのデータ投入のような、決まった手順で繰り返し行う業務はRPA化に適した業務ですよ。Webサイトでのクリック＆入力の自動化ももちろんできます。WinActorは画面上で色々な動きをする部品を組み合わせるだけで、だれでもカンタンに自動化ができますよ。」



「それなら私にもできそうです！自動化が出来れば登録の時間を別の仕事にあてることができそうですし、なにより手入力でのミスが減らせるのがありがたいです。」



「まずは手作業と、WinActorを使った自動化とで業務がどう変わるか見てみましょう。」



「WinActorを使った自動化では以下のように業務が変わります。」

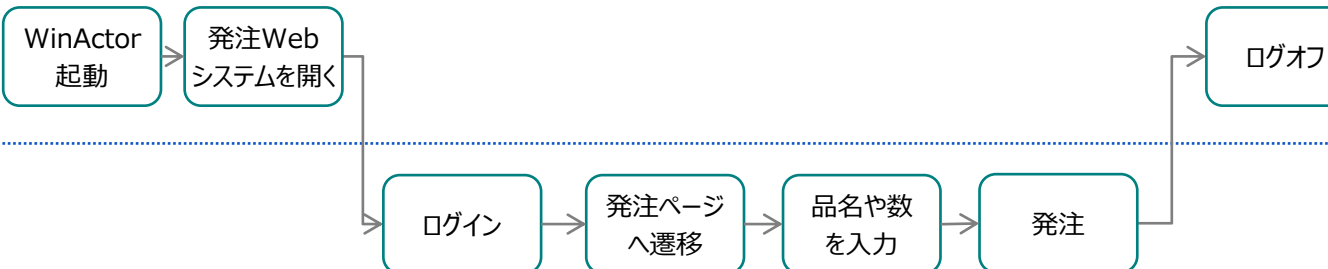
手作業

人



自動化

人



「手作業の業務がとても減りますね。私はWinActorを起動して最後にログオフを行うだけになります。」



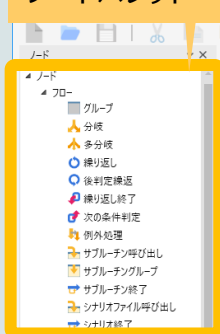
「先程、画面上で色々な動きをする部品と話しましたが、これはフローチャート上に配置することでマウスやキーボードの動きを自動化してくれる部品のことです。」



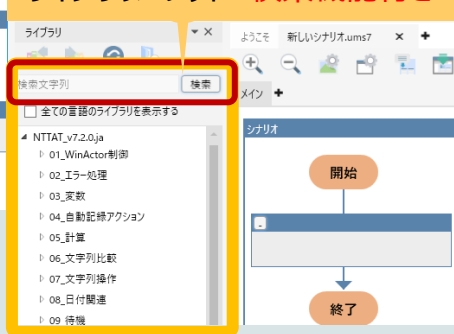
「WinActorの画面だと、どのあたりを見ればよいですか？」

「画面左上のパレットエリアに使用できる部品がカテゴリ毎に分類されています。さらに、ノードパレットとライブラリパレットに分かれています。」

ノードパレット



ライブラリパレット 検索機能付き



「色々なシナリオを作成していくと、再利用すると便利な動きがあると思います。毎回、すべて手作業でシナリオ作成するのは大変ですから、ライブラリパレットで必要な動きをする部品を検索機能も使い探せるようになると便利です！この後の操作演習でも練習しますので、気にかけておいてくださいね！」

Point!



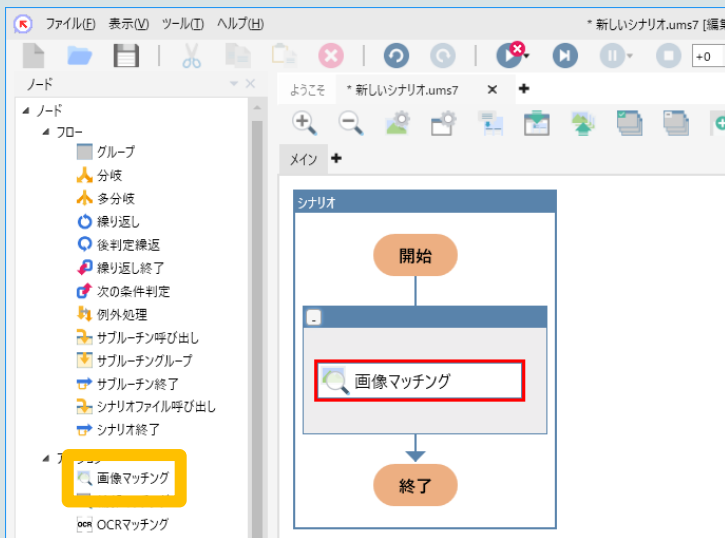


「今回のシナリオ作成でのキーポイントは、＜画像マッチング＞というノードです。」



「画像マッチング？」

「画面に表示されているブラウザーやメールソフトを画像として認識し、事前に設定した画像があればマウスをクリックなどの動作をするノードです。」

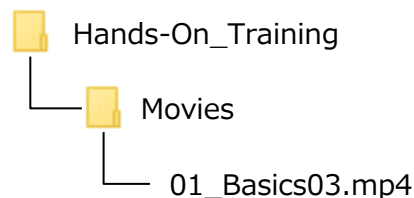


「具体的に言うと、天気予報のページを見ていて、雨のマークがあるから詳細天気予報へのリンクボタンをクリックする。といったことができるようになります。」

「画像マッチングは使いやすいノードであり、かつ色々な場面で活用できます。ぜひ使えるようになりましょう！」



「それでは実際のWinActorの動きを見てみましょう。以下フォルダのファイル名をダブルクリックして再生してください。」



ファイルが存在しない場合、本書の「はじめに」を参照し、資材のダウンロードを行ってください。



「それでは実際にシナリオを一緒に作っていきましょう！」



「説明に使うブラウザは Microsoft Edge です。手順は同じですから Google Chrome を使ってもOKです。Chromeを使いたい人は、Edge を Chrome に読み替えて進めてください。」



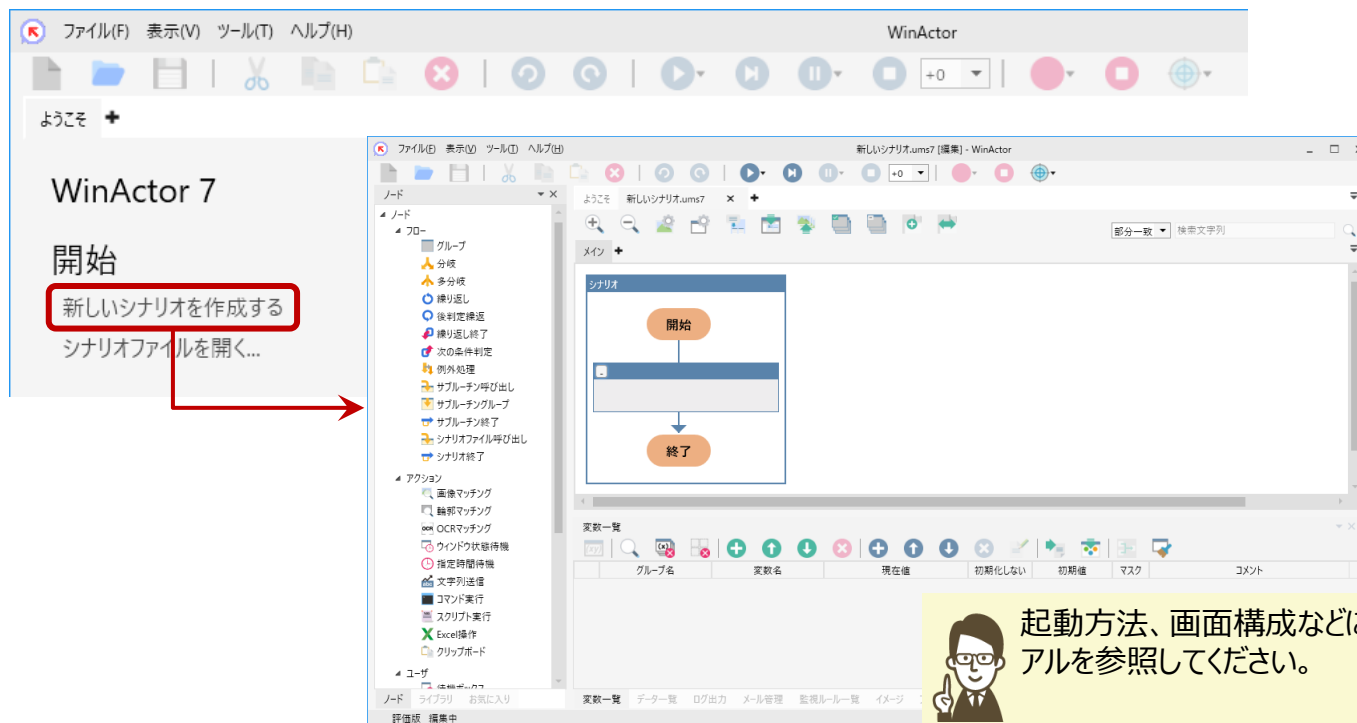
「まずはWinActorを使っでの操作説明をしますが、次スライドからの” 3. Webページのクリック&入力を自動化する”の最後に、初心者向けのWinActorの紹介があります。シナリオ作成途中で難しいと感じた場合、こちらを試してみるのも良いでしょう。」

3

Webページのクリック&入力を自動化する

準備 WinActorの起動

- 1 スタートメニューなどからWinActorを起動します。
- 2 「新しいシナリオを作成する」をクリックすると、WinActorのメイン画面が表示されます。



準備

Microsoft Edgeへ拡張機能をインストール（初回のみ）

1

＜ようこそ＞画面の＜マニュアル＞欄から
＜Chrome拡張機能インストールマニュアル＞をクリック
して、マニュアルを開きます。（＜ようこそ画面＞のレイア
ウトが違う場合は、右上のトグルボタンをクリックします。）



2

「Chrome拡張機能インストールマニュアル」の
「3. Microsoft Edgeへのインストール手順」を参照して、
拡張機能をインストールしてください。（初回のみ）



Chromeを使う人は
「2. Google Chromeへのインストール手順」
を参照してインストールしてください。

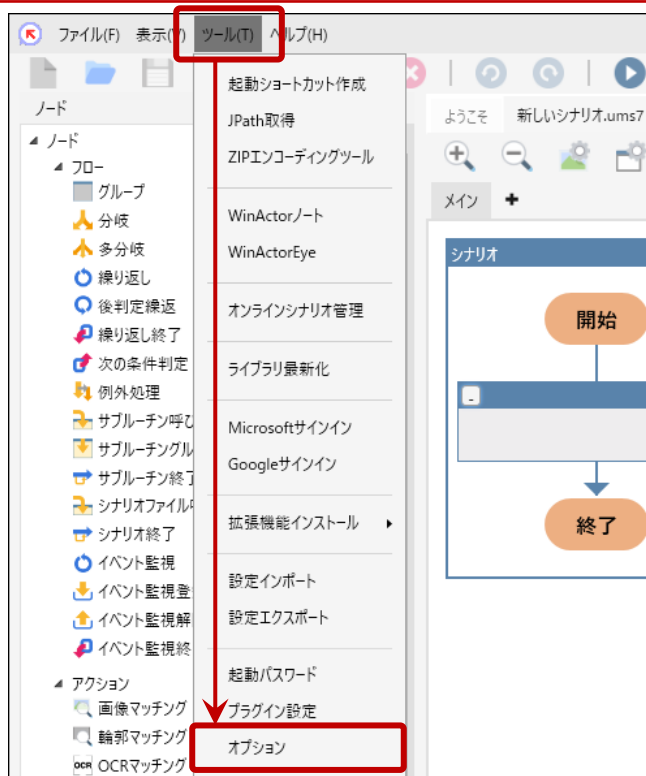
3

Webページのクリック&入力を自動化する

準備

Microsoft Edgeの拡張機能を有効化（初回のみ）

1 メニューバーの[ツール]-[オプション]を選択します。



2 オプション画面の<ブラウザ>タブをクリックし、<Edge>のプルダウンリストから<拡張機能>を選択します。（初回のみ）



3 <OK> ボタンをクリックします。



Chromeを使う人は
<Chrome>のプルダウンリストから
<拡張機能>を選択してください。

3

Webページのクリック&入力を自動化する



「これから以下の業務についてシナリオを作成していきます。」



準備 ログイン画面の表示

3-1 ユーザーIDテキストボックスへのカーソル移動

3-2 ユーザーIDテキストボックスへの入力

3-3 パスワードテキストボックスへの入力

3-4 ログインボタンのクリック

3

Webページのクリック&入力を自動化する

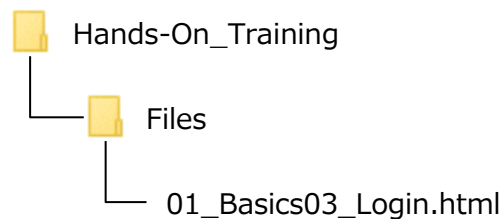
準備

ログイン画面の表示

- 1 「01_Basics03_Login.html」を右クリックして、「プログラムから開く」から「Microsoft Edge」を選択します。



使用するファイル



デモシステムを起動するブラウザーには
Edge または Chrome を使用してください。

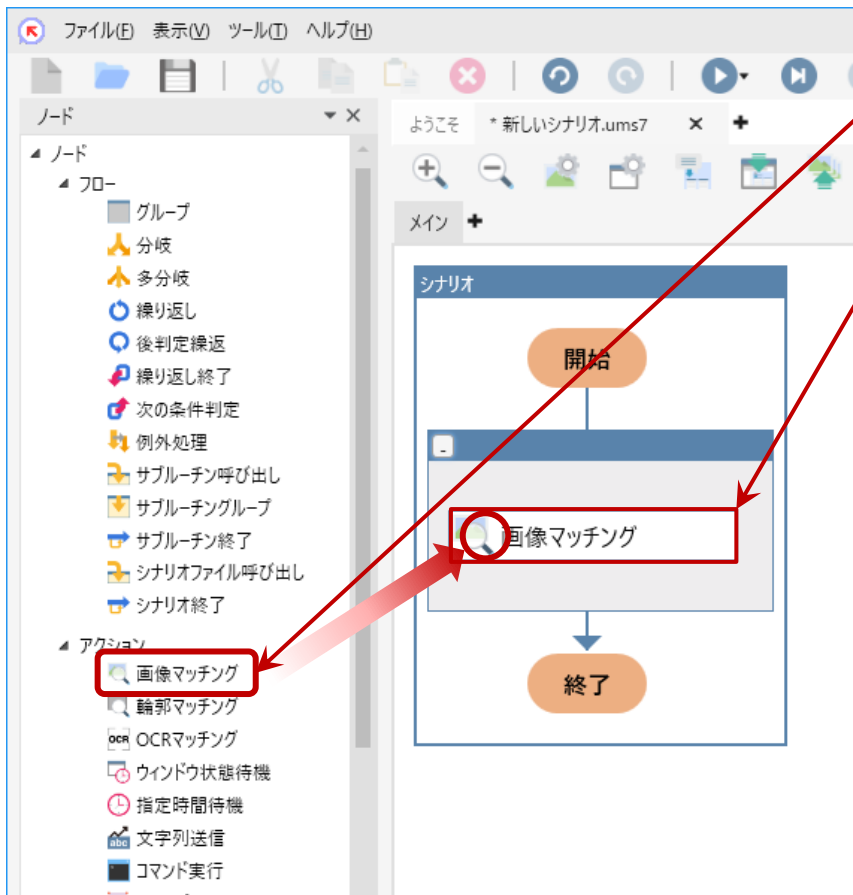


Chromeを使う人は「プログラムから開く」から
「Google Chrome」を選択してください。

3

Webページのクリック&入力を自動化する

3-1 ユーザーIDテキストボックスへのカーソル移動



1 <画像マッチング> ノードをドラッグし、シナリオ編集エリア上でドロップします。

2 配置したノードをダブルクリックします。(プロパティを表示)



カーソル移動には、<クリック> ライブラリ、<エミュレーション> ライブラリを使用する方法がありますが、今回は<画像マッチング> ノードを使用します。

3

Webページのクリック&入力を自動化する

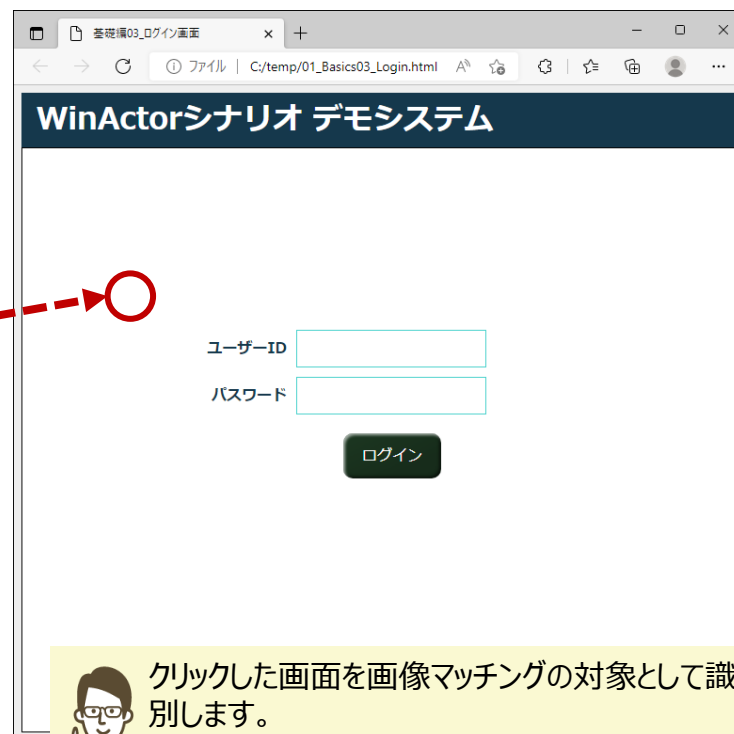
3-1 ユーザーIDテキストボックスへのカーソル移動

3 <ウィンドウ識別名>の青いスコープボタンをクリックします。



どの画面に対して動作を行うか、青いスコープボタンで指定します。

4 操作したい画面上にマウスカーソルを移動し、クリックします。



クリックした画面を画像マッチングの対象として識別します。

3

Webページのクリック&入力を自動化する

3-1 ユーザーIDテキストボックスへのカーソル移動



5 <マッチング画像> ボタンをクリックします。

さきほど取り込んだ画面から、テキストボックスのある範囲を赤枠で囲み指定します。



6



<マッチング画像> ボタンは既に選択されている場合、クリックは不要です。

3

Webページのクリック&入力を自動化する

3-1 ユーザーIDテキストボックスへのカーソル移動



7 <マウス操作位置> ボタンをクリックします。

8 さきほど取り込んだ画面から、ユーザーIDテキストボックスの位置でクリックします。

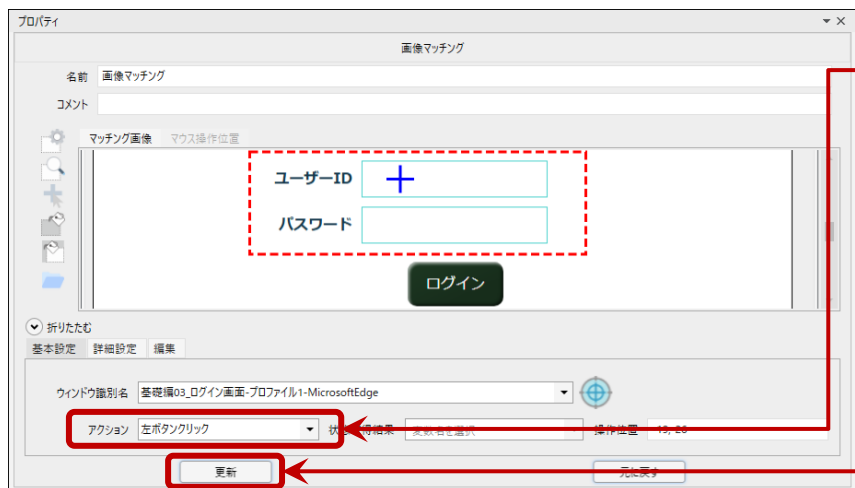


<マッチング画像> は、取り込んだ画面からアクションのターゲットとなる画像の範囲を指定し、<マウス操作位置> は、<マッチング画像> で指定した画像の中で、具体的なアクションを動作させる場所を指定します。

3

Webページのクリック&入力を自動化する

3-1 ユーザーIDテキストボックスへのカーソル移動



9

＜アクション＞のプルダウンリストから＜左ボタンクリック＞を選択します。

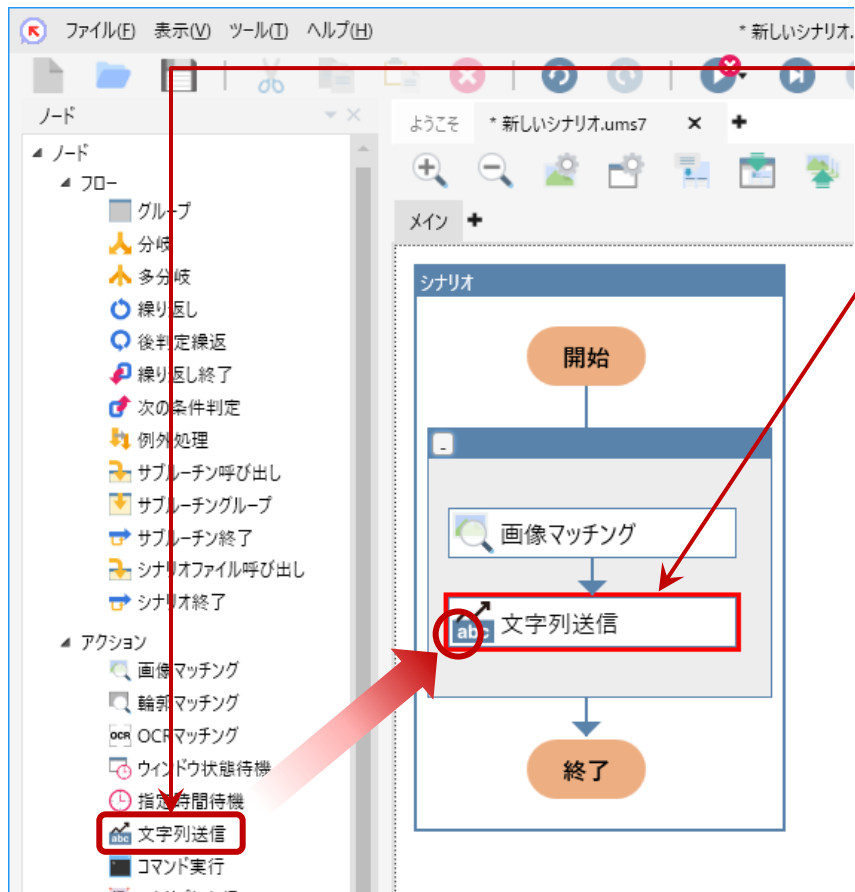
10

＜更新＞ボタンをクリックします。

3

Webページのクリック&入力を自動化する

3-2 ユーザーIDテキストボックスへの入力



1 <文字列送信> ノードをドラッグし、シナリオ編集エリア上でドロップします。

2 配置したノードをダブルクリックします。(プロパティを表示)



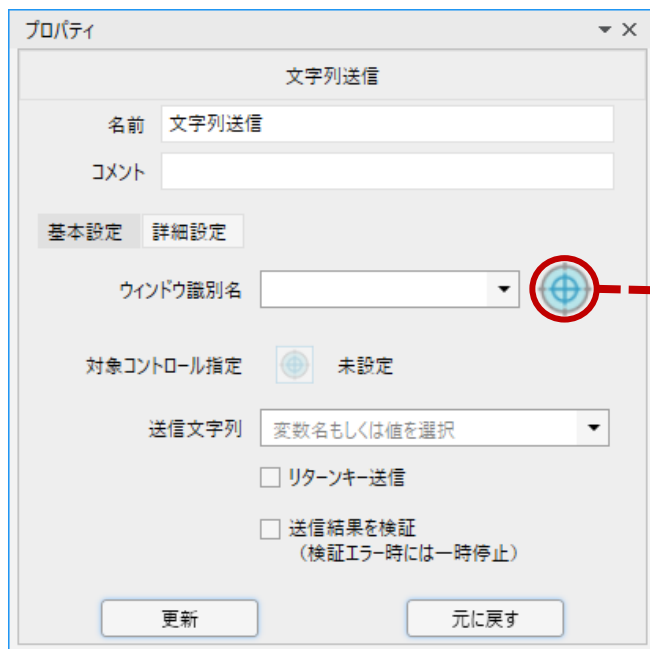
シナリオ上で文字入力させるには、いくつかの設定方法がありますが、今回は「文字列送信」を使用します。

3

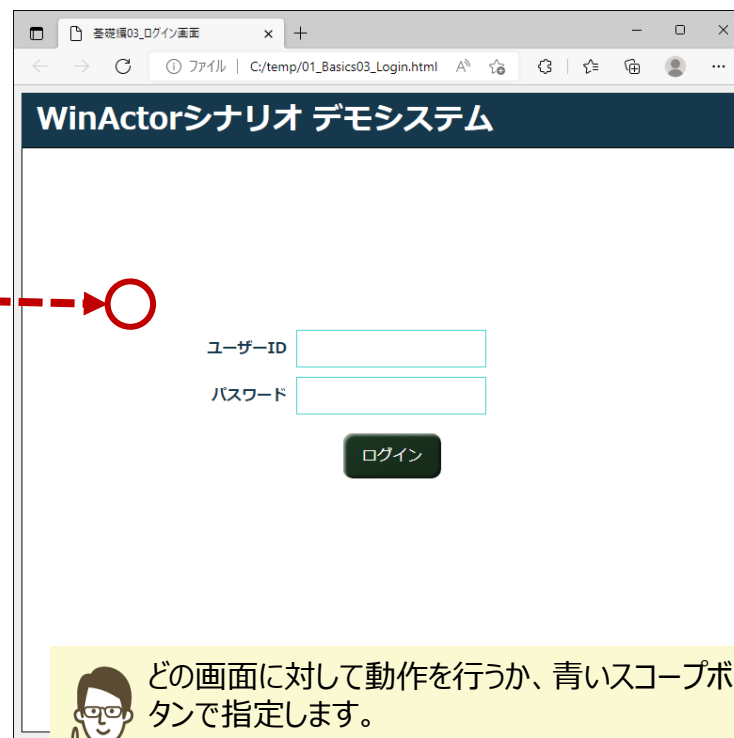
Webページのクリック&入力を自動化する

3-2 ユーザーIDテキストボックスへの入力

3 <ウィンドウ識別名>の青いスコープボタンをクリックします。



4 操作したい画面上にマウスカーソルを移動し、クリックします。



3

Webページのクリック&入力を自動化する

3-2 ユーザーIDテキストボックスへの入力

The screenshot shows the 'Properties' (プロパティ) dialog box for a 'Text String' (文字列送信) action. The 'Basic Settings' (基本設定) tab is active. The 'Name' (名前) is 'Text String' (文字列送信). The 'Window Identification Name' (ウィンドウ識別名) is 'Basic Edit 03 Login Screen' (基礎編03_ログイン画面). The 'Target Control Specification' (対象コントロール指定) is 'Not Set' (未設定). The 'Send String' (送信文字列) dropdown is set to 'Select variable name or value' (変数名もしくは値を選択). The 'Update' (更新) button is highlighted with a red box. Red arrows point from the 'Update' button to the 'Send String' dropdown and from the 'Update' button to the instruction box for step 6.

5 <送信文字列> に、「値⇒ User01」を入力します。

6 <更新> ボタンをクリックします。

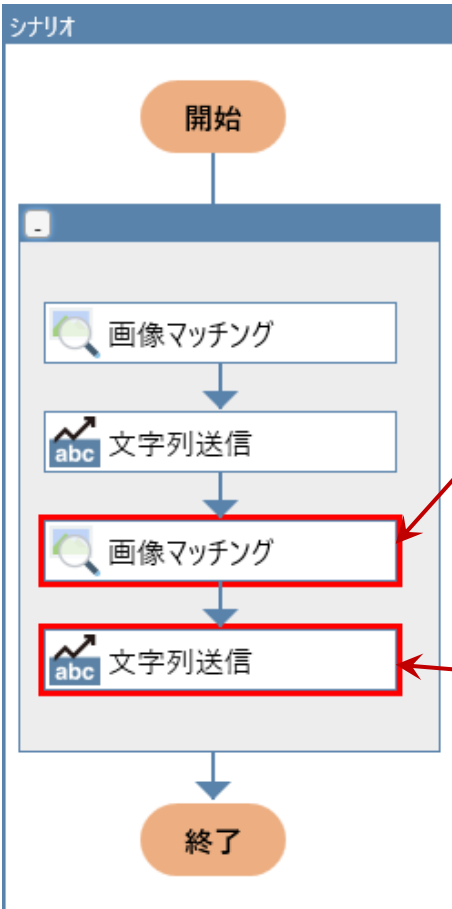


<送信文字列> に入力した値が、実際に入力される文字列になります。

3

Webページのクリック&入力を自動化する

3-3 パスワードテキストボックスへの入力



1 画像マッチングについては 3-1、
文字列送信については 3-2と同じ手順で設定してください。

1.1 画像マッチングは、＜パスワード＞のテキストボックスを指定します。



1.2 文字列送信は、「Password」を入力します。

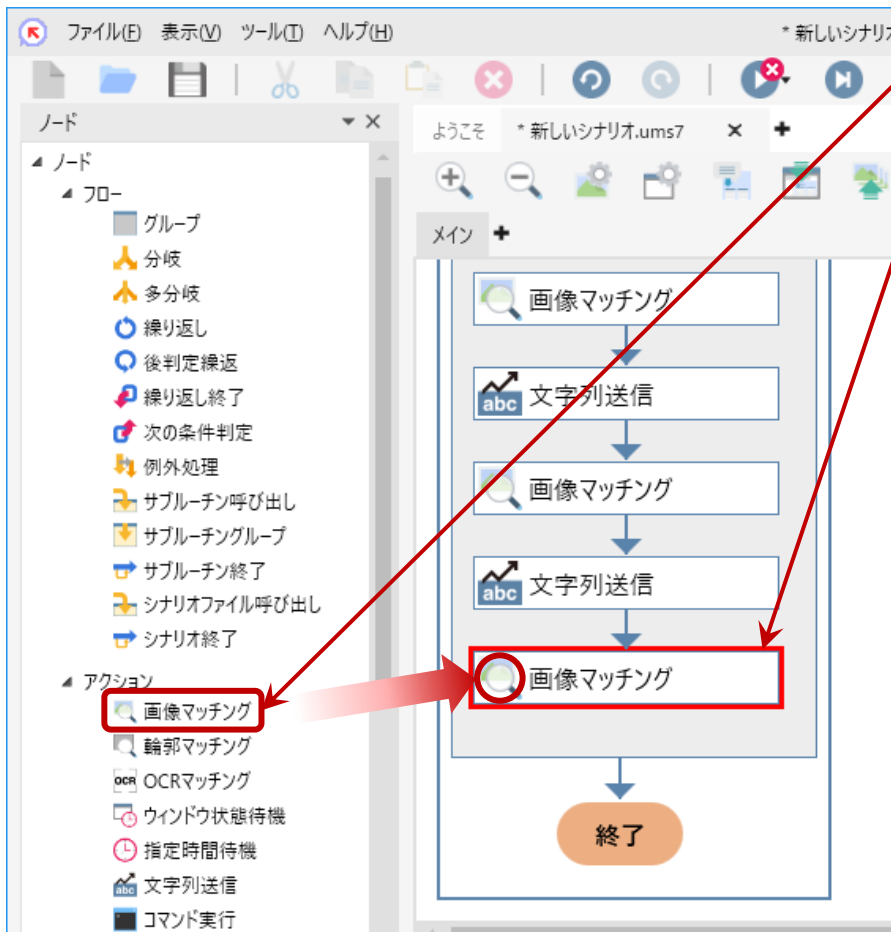


上記の設定で、
ユーザID : User01
パスワード : Password
を入力する動作が完成します。

3

Webページのクリック&入力を自動化する

3-4 ログインボタンのクリック



1 <画像マッチング> ノードをドラッグし、シナリオ編集エリア上でドロップします。

2 配置したノードをダブルクリックします。(プロパティを表示)



<画像マッチング> はテキストボックスへのクリック以外にも、ログインボタンなどボタンのクリックにも利用できます。

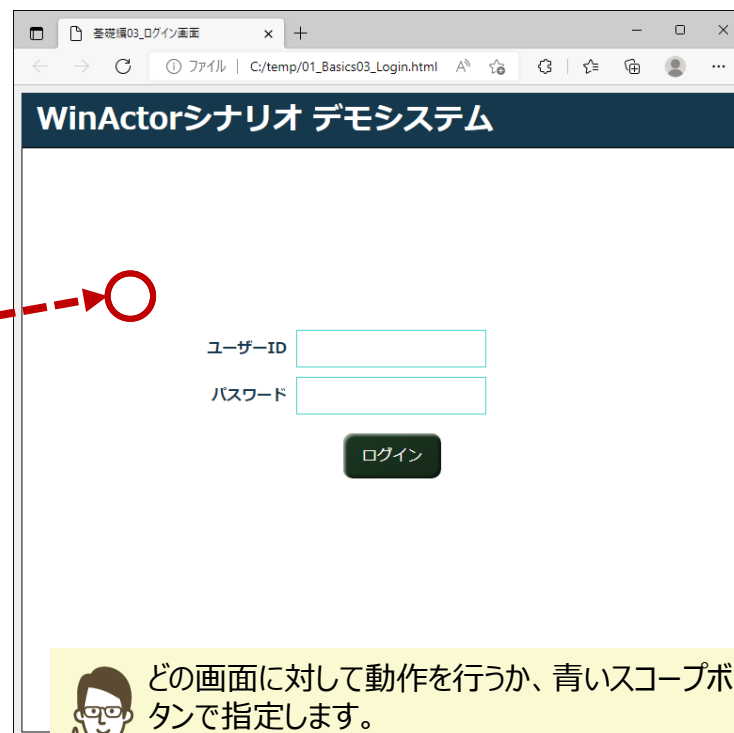
3

Webページのクリック&入力を自動化する

3-4 ログインボタンのクリック

3 <ウィンドウ識別名>の青いスコープボタンをクリックします。

4 操作したい画面上にマウスカーソルを移動し、クリックします。



どの画面に対して動作を行うか、青いスコープボタンで指定します。

3

Webページのクリック&入力を自動化する

3-4 ログインボタンのクリック



5 <マッチング画像> ボタンをクリックします。

6 さきほど取り込んだ画面から、<ログイン> ボタンのある範囲を指定します。

3

Webページのクリック&入力を自動化する

3-4 ログインボタンのクリック



7 <マウス操作位置> ボタンをクリックします。

8 さきほど取り込んだ画面から、<ログイン> ボタンの位置でクリックします。



今回はログインボタンをクリックすることが目的であるため、<マウス操作位置> はログインボタンになります。

3

Webページのクリック&入力を自動化する

3-4

ログインボタンのクリック

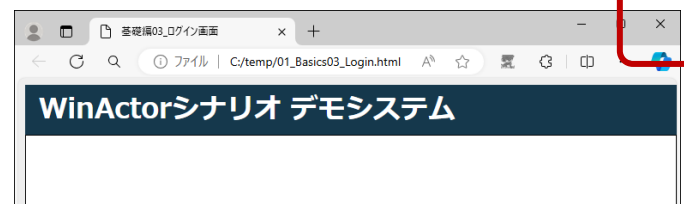


9 <アクション> のプルダウンリストから <左ボタンクリック> を選択します。

10 <更新> ボタンをクリックします。

Edgeの <閉じる> ボタンをクリックして終了させます。

11

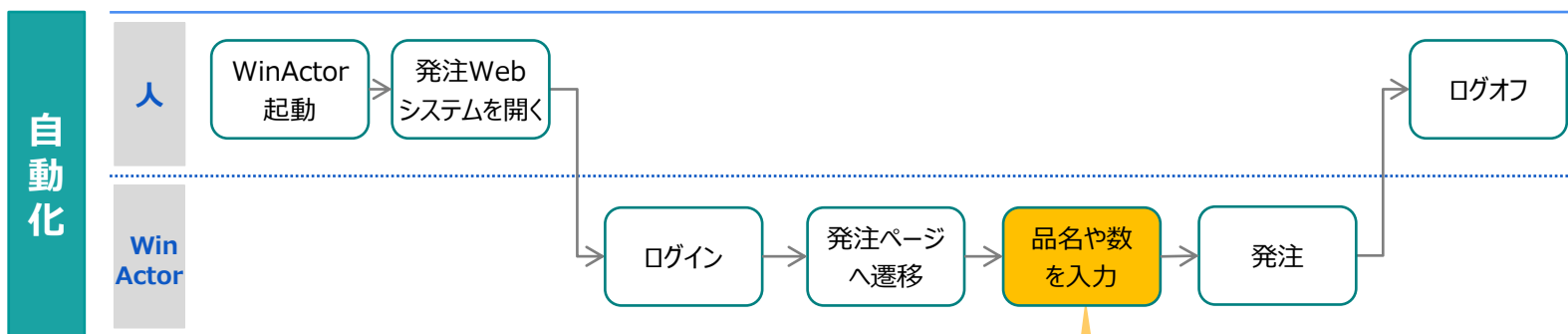


3

Webページのクリック&入力を自動化する



「これから以下の業務についてシナリオを作成していきます。」



3-5 発注登録画面の表示

3-6 注文日の入力

3-7 仕入先コード・担当者コードの入力

3-8 商品コードの選択

3

Webページのクリック&入力を自動化する

3-5

発注登録画面の表示

- 1 「02_Basics03_Hacchu_Touroku.html」を右クリックして、「プログラムから開く」から「Microsoft Edge」を選択します。

WinActorシナリオ デモシステム

メニュー 発注登録 発注一覧 在庫登録 支払登録

発注登録

注文日 2019/12/9

仕入先コード 123456 仕入先名

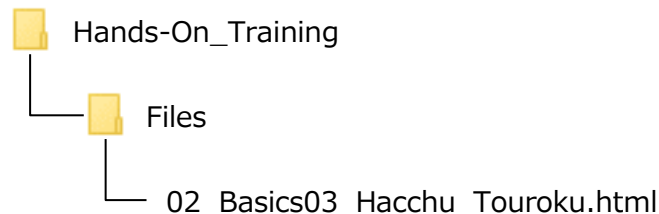
担当者コード WA011 担当者名

商品コード	商品名	数量	金額
未選択			
未選択			
未選択			

保存 発注

小計	
消費税(10%)	
合計金額	

使用するファイル



デモシステムを起動するブラウザーには
Edge または Chrome を使用してください。



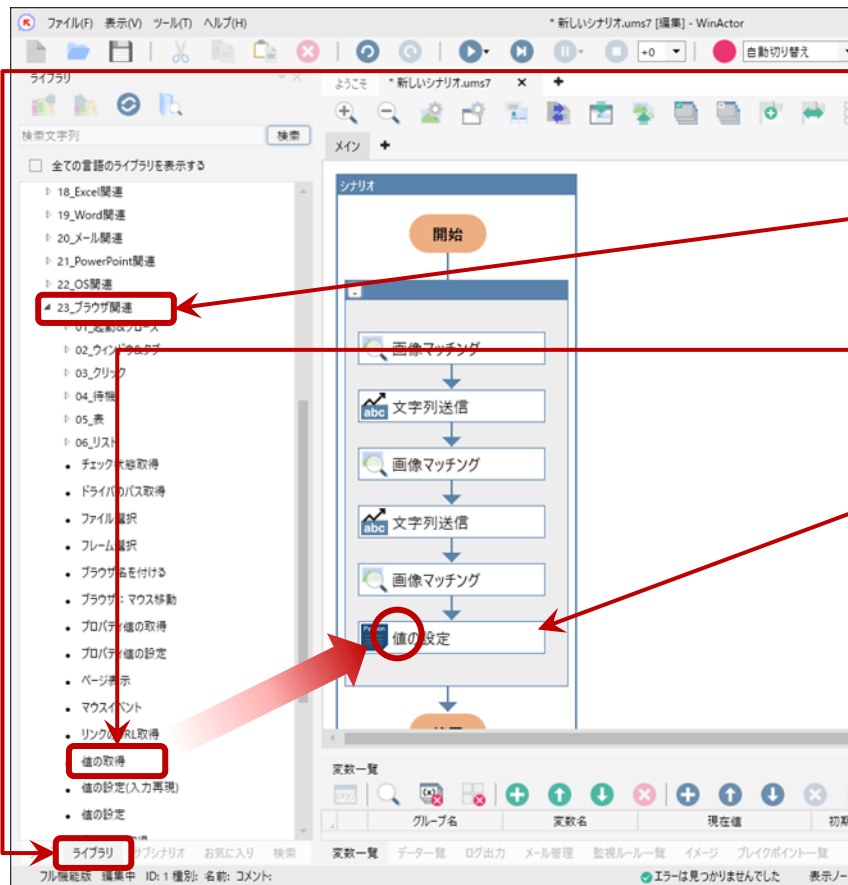
Chromeを使う人は「プログラムから開く」から
「Google Chrome」を選択してください。

3

Webページのクリック&入力を自動化する

3-6

注文日の入力



1 <ライブラリ> をクリックします。

2 <23_ブラウザ関連> をダブルクリックして、リストを展開します。

3 <値の設定> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

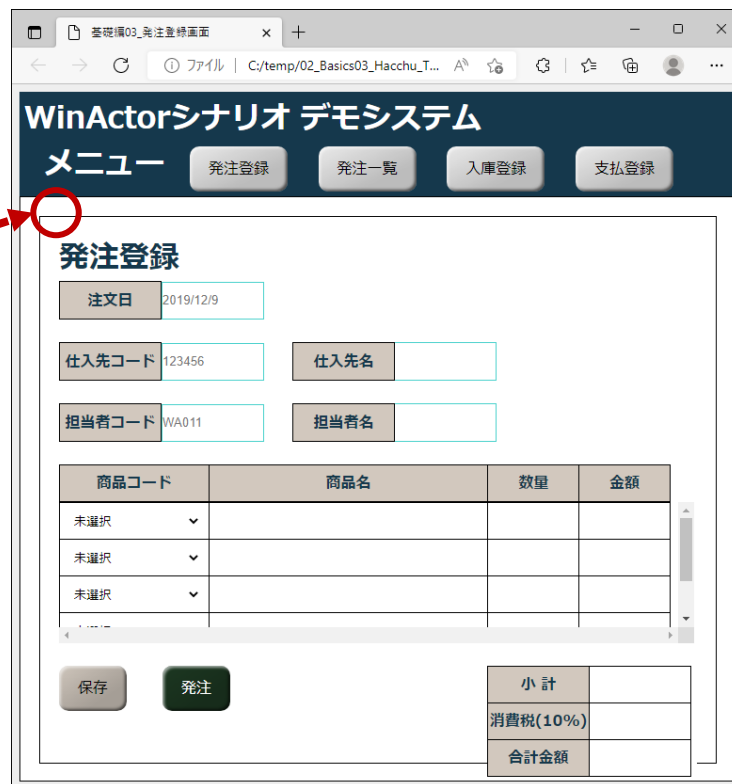
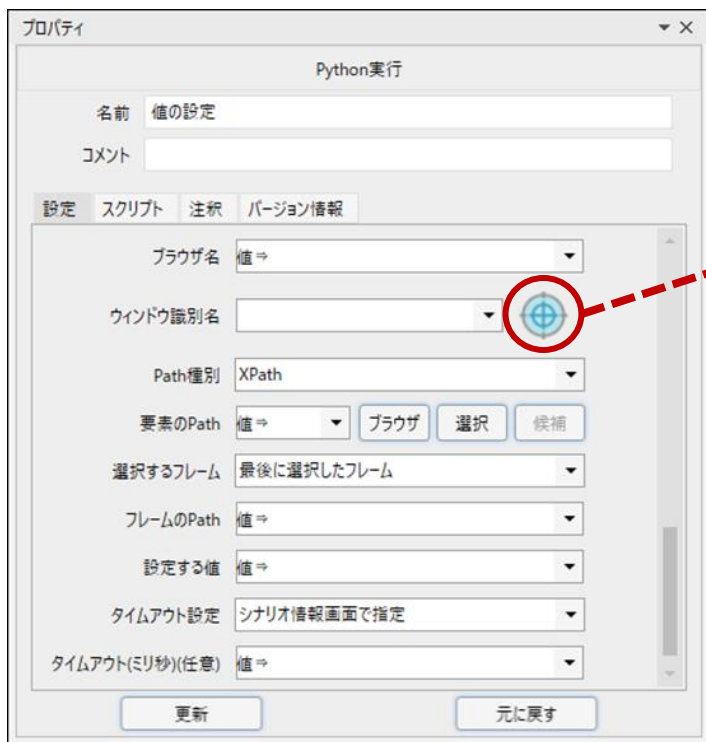
3

Webページのクリック&入力を自動化する

3-6 注文日の入力

5 <ウィンドウ識別名>の青いスコープボタンをクリックします。

6 操作したい画面上にマウスカーソルを移動し、クリックします。



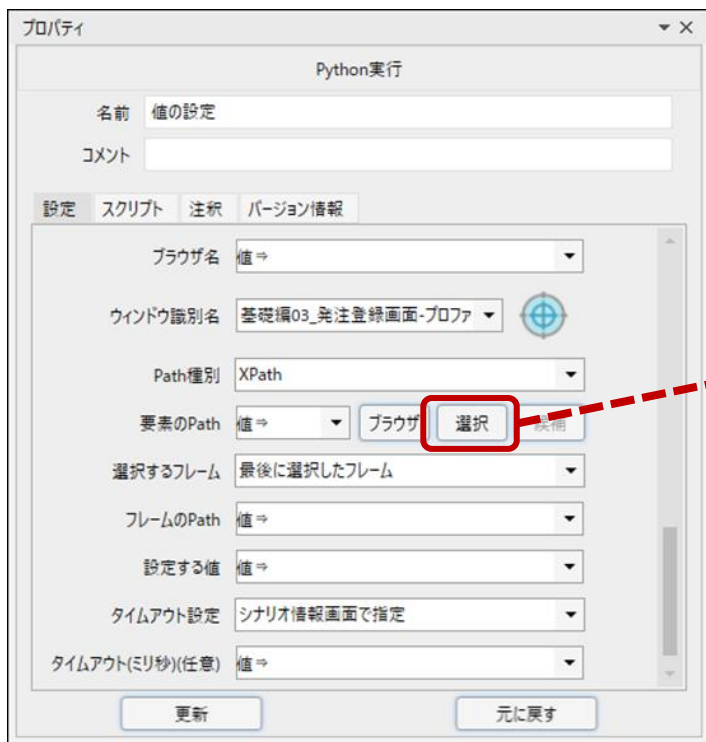
3

Webページのクリック&入力を自動化する

3-6 注文日の入力

7 <要素のPath> の<選択> ボタンをクリックします。

8 <注文日> テキストボックスにマウスカーソルを移動し、色が変わったら、クリックします。



3

Webページのクリック&入力を自動化する

3-6

注文日の入力

プロパティ

Python実行

名前 値の設定

コメント

設定 スクリプト 注釈 バージョン情報

ブラウザ名 値⇒

ウィンドウ識別名 基礎編03_発注登録画面-プロファ

Path種別 XPath

要素のPath 値⇒ //*[@ ブラウザ 選択 候補

選択するフレーム トップフレーム

フレームのPath 値⇒

設定する値 値⇒ 2019/12/20

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値⇒

更新 元に戻す

9

＜設定する値＞に、「値⇒ 2019/12/20」を入力します。

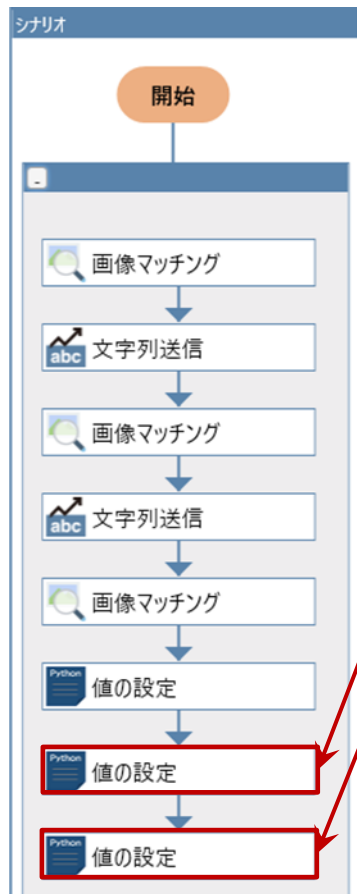
10

＜更新＞ボタンをクリックします。

3

Webページのクリック&入力を自動化する

3-7 仕入先コード・担当者コードの入力



1 3-6と同じ手順で設定してください。

1.1 <仕入先コード> に、「123456」と入力します。

1.2 <担当者コード> に、「WA011」と入力します。

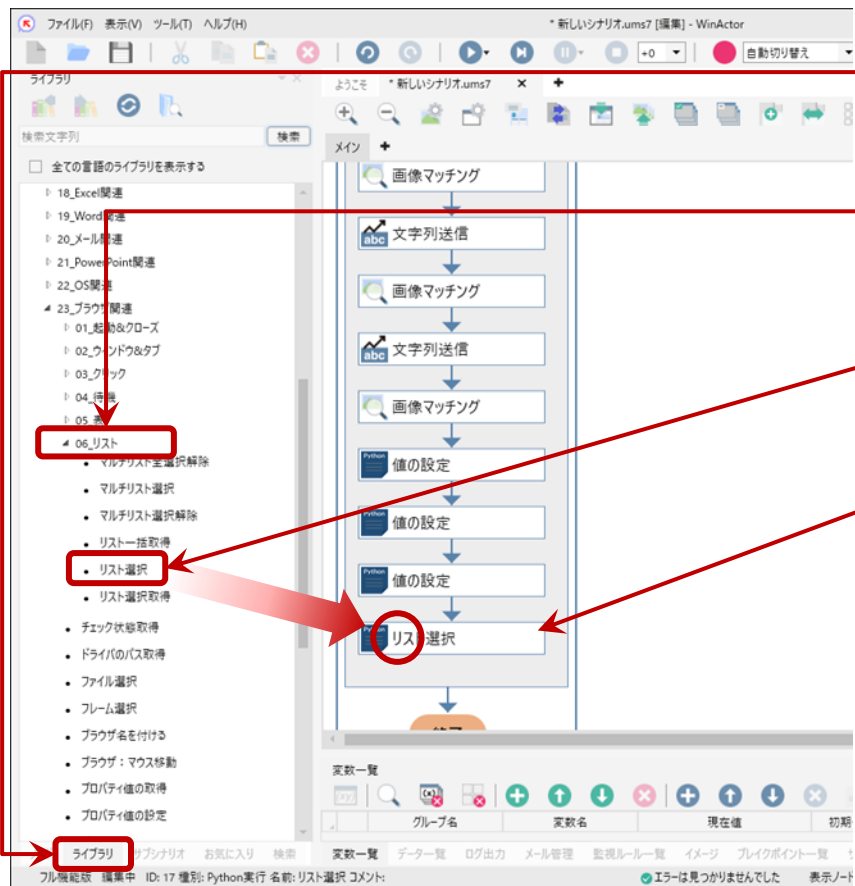


上記の設定で、注文日、仕入先コード、担当者コードを入力する動作が完成します。

3

Webページのクリック&入力を自動化する

3-8 商品コードの選択



1 <ライブラリ> をクリックします。

2 <06_リスト> をダブルクリックして、リストを展開します。

3 <リスト選択> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

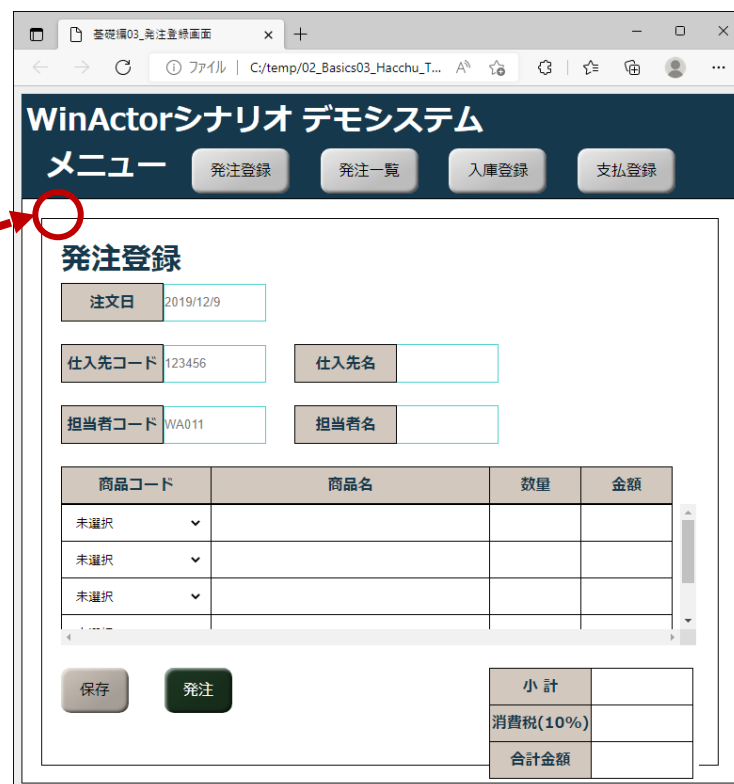
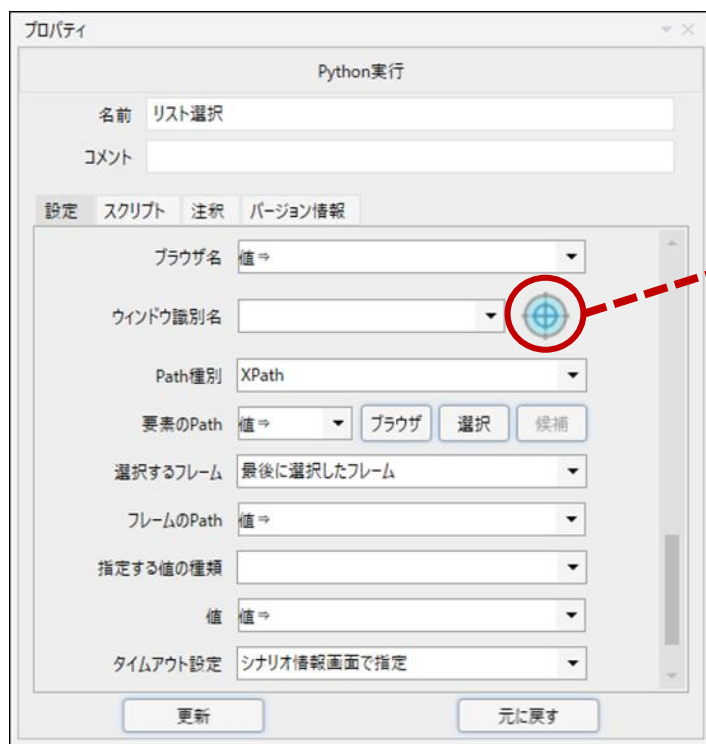
3

Webページのクリック&入力を自動化する

3-8 商品コードの選択

5 <ウィンドウ識別名>の青いスコープボタンをクリックします。

6 操作したい画面上にマウスカーソルを移動し、クリックします。

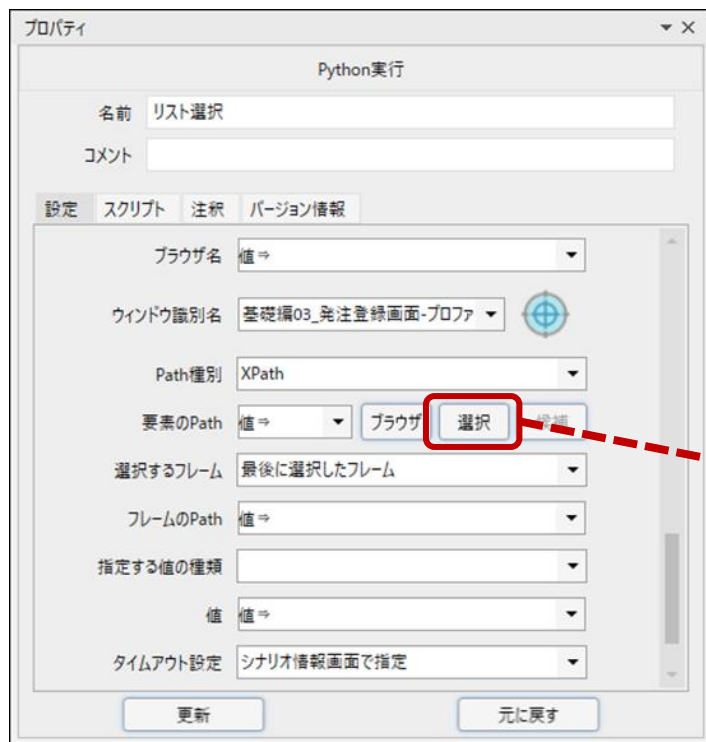


3

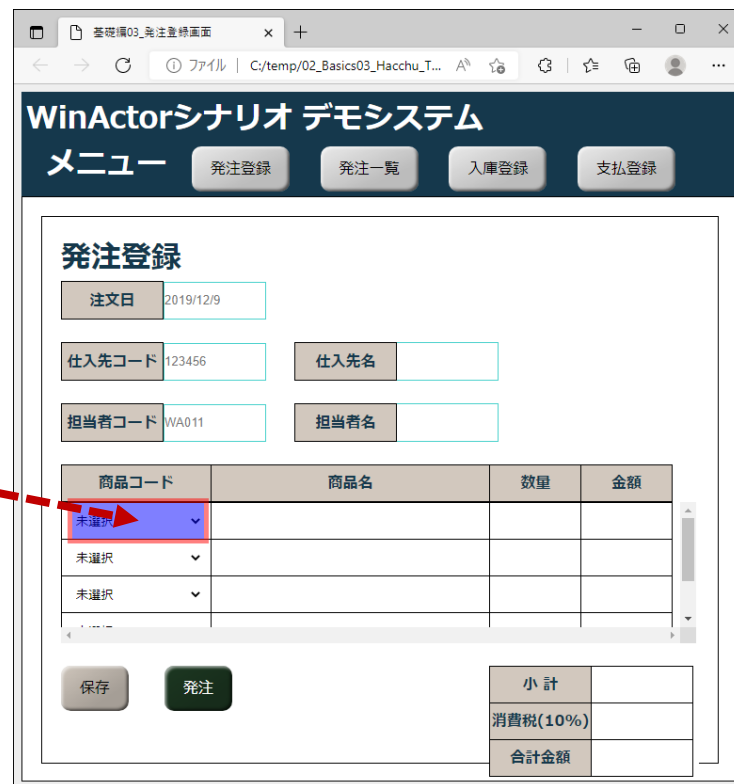
Webページのクリック&入力を自動化する

3-8 商品コードの選択

7 <要素Path> の<選択> ボタンをクリックします。



8 <商品コード> リストにマウスカーソルを移動し、色が変わったら、クリックします。



3

Webページのクリック&入力を自動化する

3-8

商品コードの選択

プロパティ Python実行

名前 リスト選択

コメント

設定 スクリプト 注釈 バージョン情報

ブラウザ名 値⇒

ウィンドウ識別名 基礎編03_発注登録画面-プロファ

Path種別 XPath

要素のPath 値⇒ //*[@ ブラウザ 選択 候補

選択するフレーム トップフレーム

フレームのPath 値⇒

指定する値の種類 value

値 値⇒ P001

タイムアウト設定 シナリオ情報画面で指定

更新 元に戻す

9

<指定する値の種類> に、「value」
<値> に、「値⇒ P001」
を入力します。

10

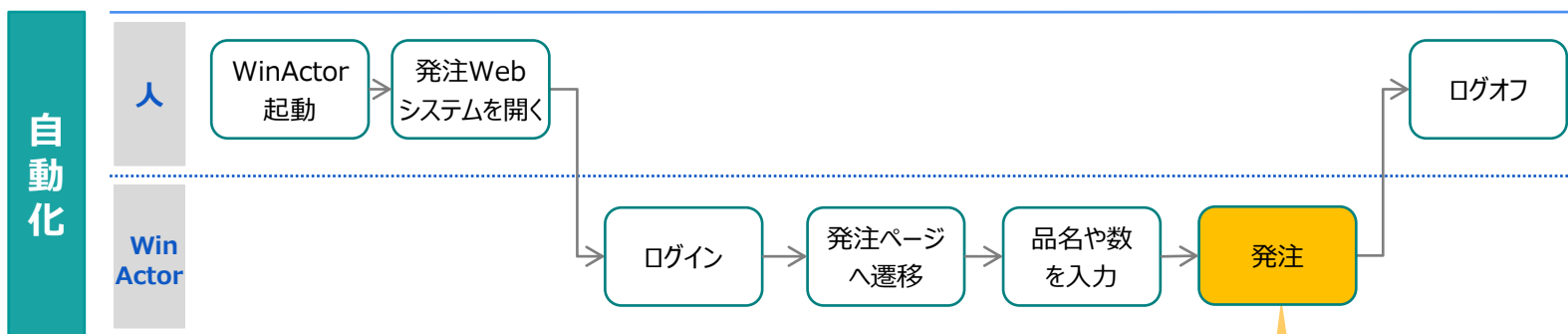
<更新> ボタンをクリックします。

3

Webページのクリック&入力を自動化する



「これから以下の業務についてシナリオを作成していきます。」



3-9

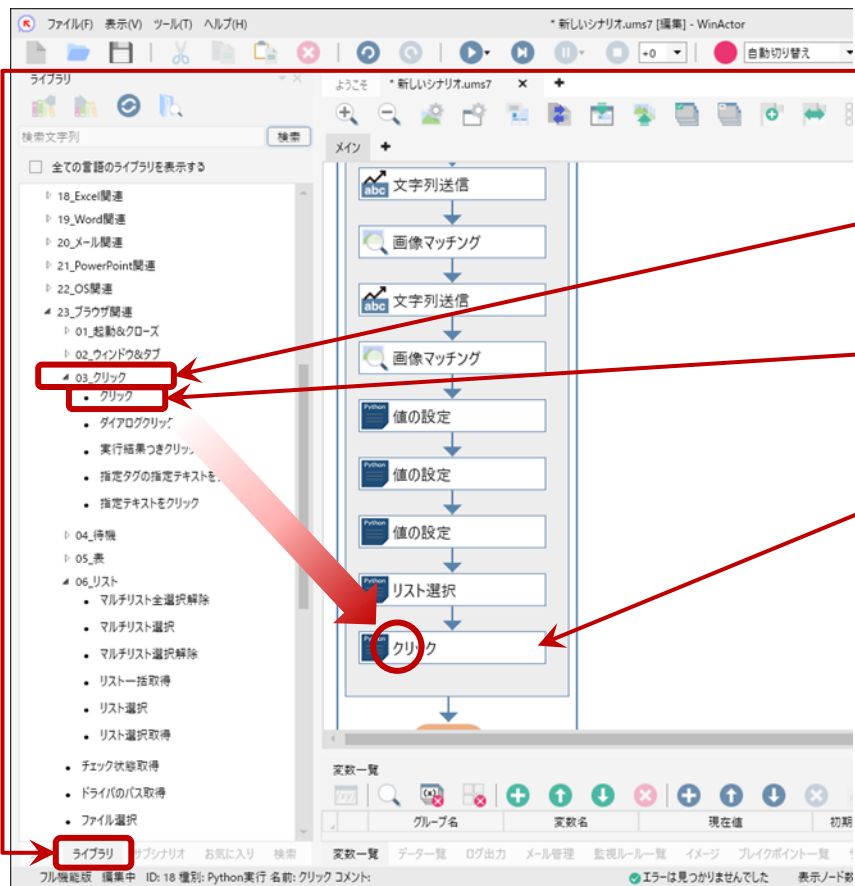
発注ボタンのクリック

3

Webページのクリック&入力を自動化する

3-9

発注ボタンのクリック



1 <ライブラリ> をクリックします。

2 <03_クリック> をダブルクリックして、リストを展開します。

3 <クリック> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

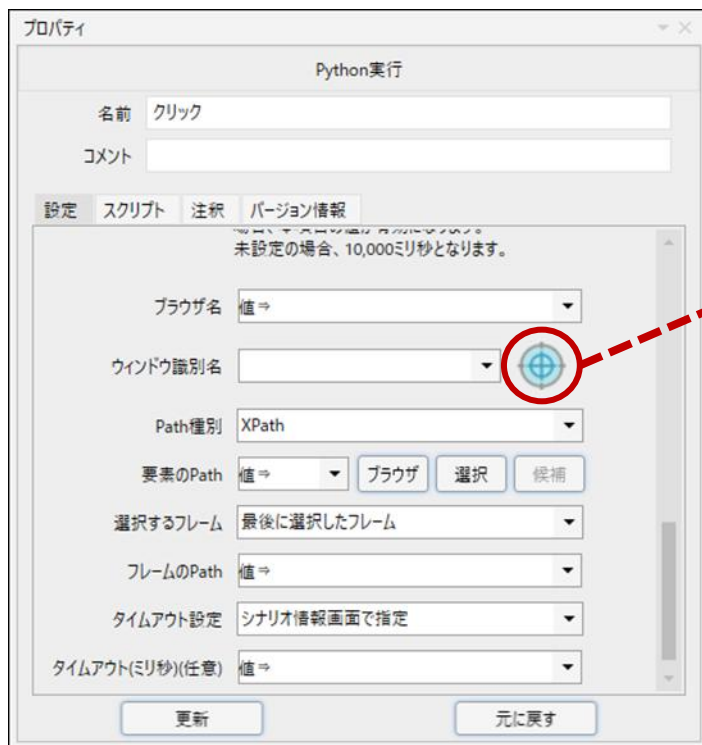
4 配置したライブラリをダブルクリックします。(プロパティを表示)

3

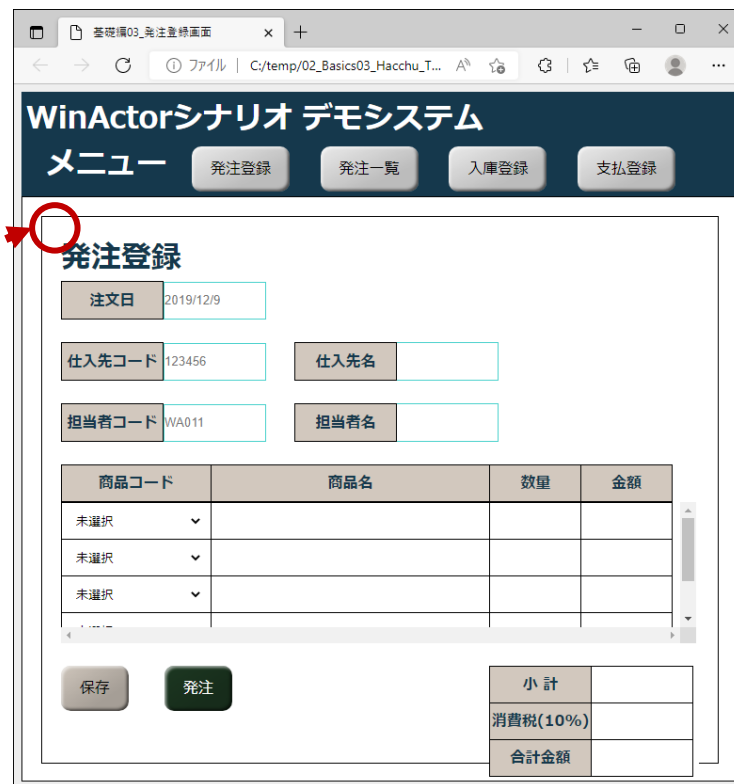
Webページのクリック&入力を自動化する

3-9 発注ボタンのクリック

5 <ウィンドウ識別名>の青いスコープボタンをクリックします。



6 操作したい画面上にマウスカーソルを移動し、クリックします。



3

Webページのクリック&入力を自動化する

3-9 発注ボタンのクリック

7 <要素のPath> の<選択> ボタンをクリックします。

プロパティ

Python実行

名前 クリック

コメント

設定 スクリプト 注釈 バージョン情報

未設定の場合、10,000ミリ秒となります。

ブラウザ名 値⇒

ウィンドウ識別名 基礎欄03_発注登録画面-プロパティ

Path種別 XPath

要素のPath 値⇒ ブラウザ **選択** 挿入

選択するフレーム 最後に選択したフレーム

フレームのPath 値⇒

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値⇒

更新 元に戻す

8 <発注> ボタンにマウスカーソルを移動し、色が変わったら、クリックします。

WinActorシナリオ デモシステム

メニュー 発注登録 発注一覧 入庫登録 支払登録

発注登録

注文日 2019/12/9

仕入先コード 123456 仕入先名

担当者コード WAO11 担当者名

商品コード	商品名	数量	金額
未選択			
未選択			
未選択			

保存 発注

小計	
消費税(10%)	
合計金額	

3

Webページのクリック&入力を自動化する

3-9

発注ボタンのクリック

9 <更新> ボタンをクリックします。

プロパティ Python実行

名前 クリック

コメント

設定 スクリプト 注釈 バージョン情報

未設定の場合、10,000ミリ秒となります。

ブラウザ名 値⇒

ウィンドウ識別名 基礎編03_発注登録画面-プロフ: 

Path種別 XPath

要素のPath 値⇒ //*[ ブラウザ 選択 候補

選択するフレーム トップフレーム

フレームのPath 値⇒

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値⇒

更新 元に戻す

3

Webページのクリック&入力を自動化する

3-10 シナリオ全体の確認

1 作成したシナリオが以下の通りになっていることを確認します。



3

Webページのクリック&入力を自動化する

3-11 シナリオの実行



1 <シナリオ実行> ボタンをクリックし、エラーなくシナリオが進むか確認します。

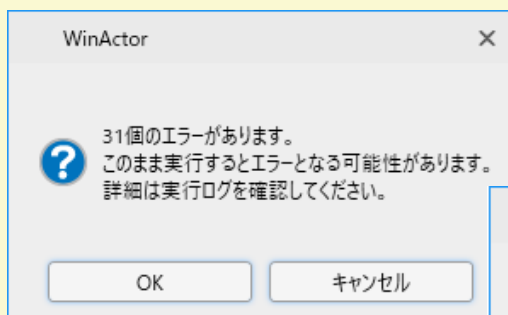
2 エラーが発生したら、エラーメッセージを確認しプロパティ等の設定を見直します。

3-11 シナリオの実行

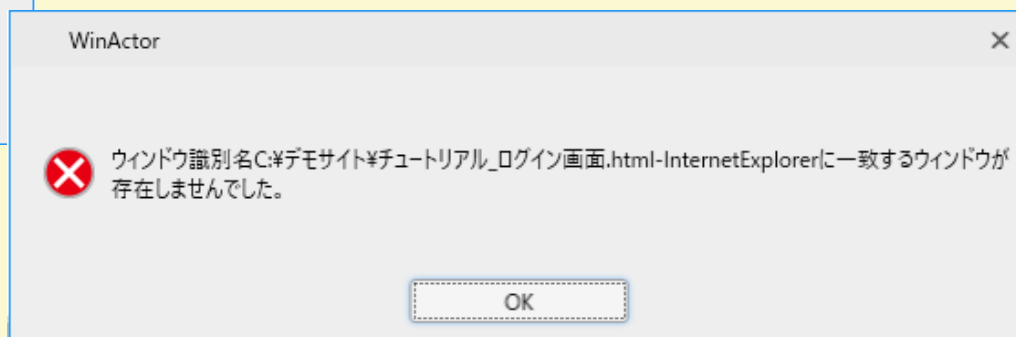


エラーメッセージについて

エラーがあるまま <シナリオ実行> ボタンをクリックすると以下のようなシナリオ実行確認ダイアログが表示されます。作ったシナリオが壊れたりすることはありませんので、動きを確認したい時は「OK」をクリックして大丈夫です。エラーのある箇所までシナリオが進むとまた以下のようなダイアログが表示されます。メッセージを確認し、エラー原因を探る手掛かりにしましょう。



シナリオ実行確認ダイアログ

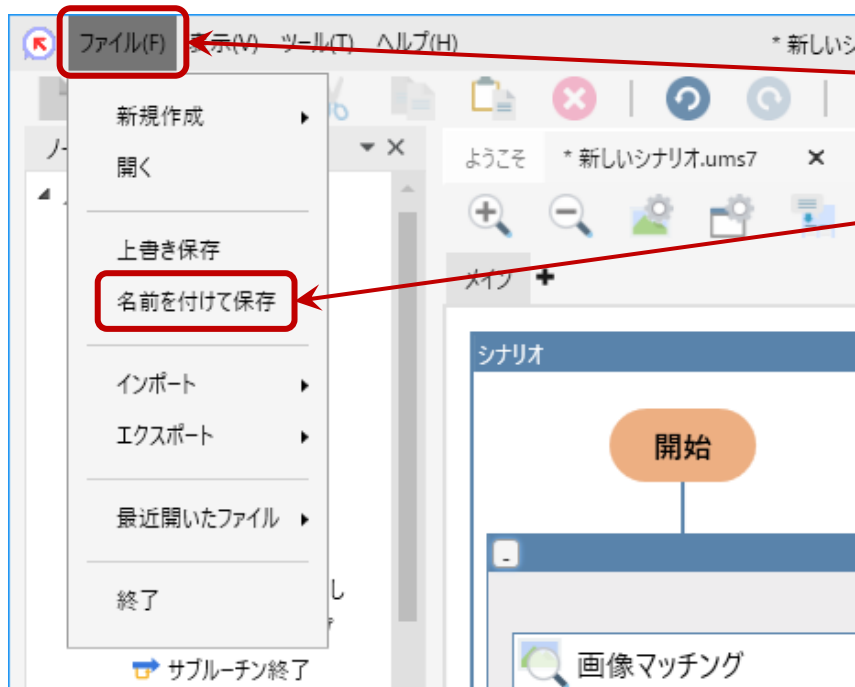


エラーダイアログ

3

Webページのクリック&入力を自動化する

3-12 シナリオの保存



1 <ファイル> メニューをクリックします。

2 <名前を付けて保存> メニューをクリックします。

3 保存先を選択し、ファイル名を付けて保存します。



すでに保存されているファイルについては、<上書き保存> から保存することができます。



「どうですか？ 宇院さんの困りごとの解決策として使えそうな気がしてきませんか？」



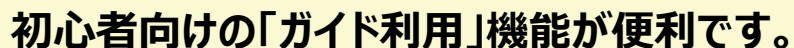
「そうですね、私の手作業がそのまま自動化されていて、手作業の軽減や誤入力も減りそうです！」



「いいですね、その調子でシナリオ作成を進めていきましょう。」

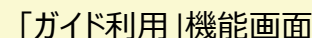
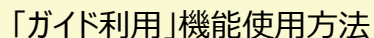
初心者向け「ガイド利用」機能のご紹介

シナリオ作りが難しかったら…

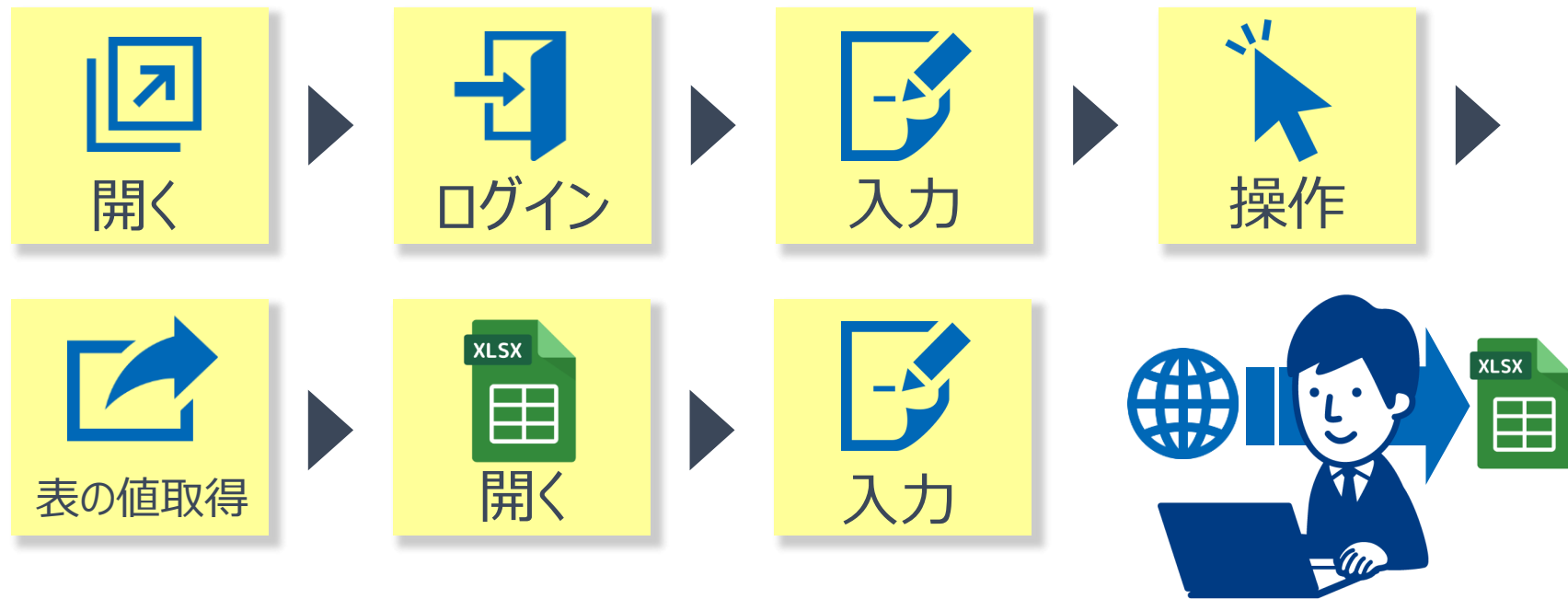


「ガイド利用」機能はWinActorのメニューバーの
ファイル > 新規作成 > 新規作成（ガイド利用）
から使用可能です。

詳細は「WinActor 操作マニュアル ガイド利用でシナリオ作成」をご覧ください。



基礎編



WebページからExcel台帳への転記を自動化する

導入

宇院さんは無事に発注システムへの登録を自動化でき、ひと安心しました。

ですが今度は、発注した品物の発注ステータスを自社の入荷台帳に転記することが手間になってきました。



「発注は無事に自動化できたけれども、今度は発注後のステータスを毎回表示して確認することが手間になってきまして…」



「WebページからExcelへの転記作業の自動化もWinActorの得意分野です。今回も既存の部品を組み合わせるだけで、カンタンに自動化が出来ますよ。」



「WinActorはExcelも利用できるんですね、Excelの台帳に直接ステータスを書き込めるとすごく助かります！」



「今回も手作業とWinActorを使った自動化とで業務がどう変わるか確認しましょう。」

4

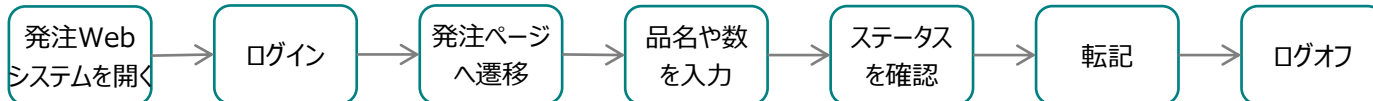
シナリオ作成 ～基礎編.2～



「WinActorを使った自動化では以下のように業務が変わります。」

手
作
業

人

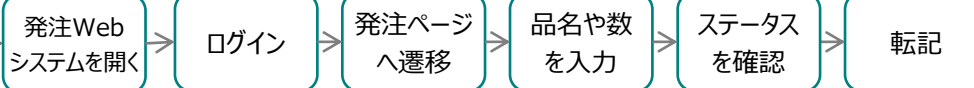


自
動
化

人

WinActor
起動

Win
Actor



「この場合も手作業の業務がととも減りますね。」



「今回は、＜表の値取得＞というノードを利用します。」



「表を取り込むんですね！」

「そのとおりです。読んで字のごとく、**ブラウザーで表示されている表を取得するノード**です。」

通番	注文日	承認日	依頼部署	依頼者氏名
1	2019/12/25	2019/12/25	経理部	鈴木太郎
2	2019/12/25	2019/12/25	経理部	田村恵一
3				
4				
5				
6				

Point!

「＜表の値取得＞を利用できるようになると、インターネット上の株価一覧や商品一覧、天気など表で表現されているコンテンツを取得できるようになり、WinActorの活用の幅が広がります。使ってみましょう！」



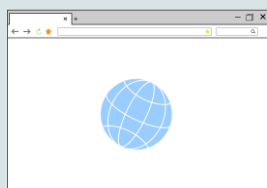


「今回はさらに、＜Excel操作＞というノードも利用します。」



「WinActorでExcelが使えるようになるんですね！」

「その通りです。前回同様読んで字のごとく、Excelを操作するノードです。このノードではExcelファイルへの書き込み、Excelファイルの読み込み、Excelマクロの実行ができます。」

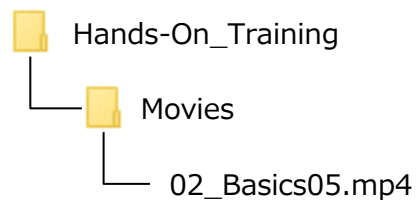


「WinActorでExcelを操作する方法は他にもありますが、今回はExcel操作ノードを利用して、Excelファイルへの書き込みを試みましょう！」

「Excelを利用できるとさらにWinActorの活用範囲がぐっと広がりますよ！」



「それでは実際のWinActorの動きを見てみましょう。以下フォルダのファイル名をダブルクリックして再生してください。」



「それでは実際にシナリオを一緒に作っていきましょう！」

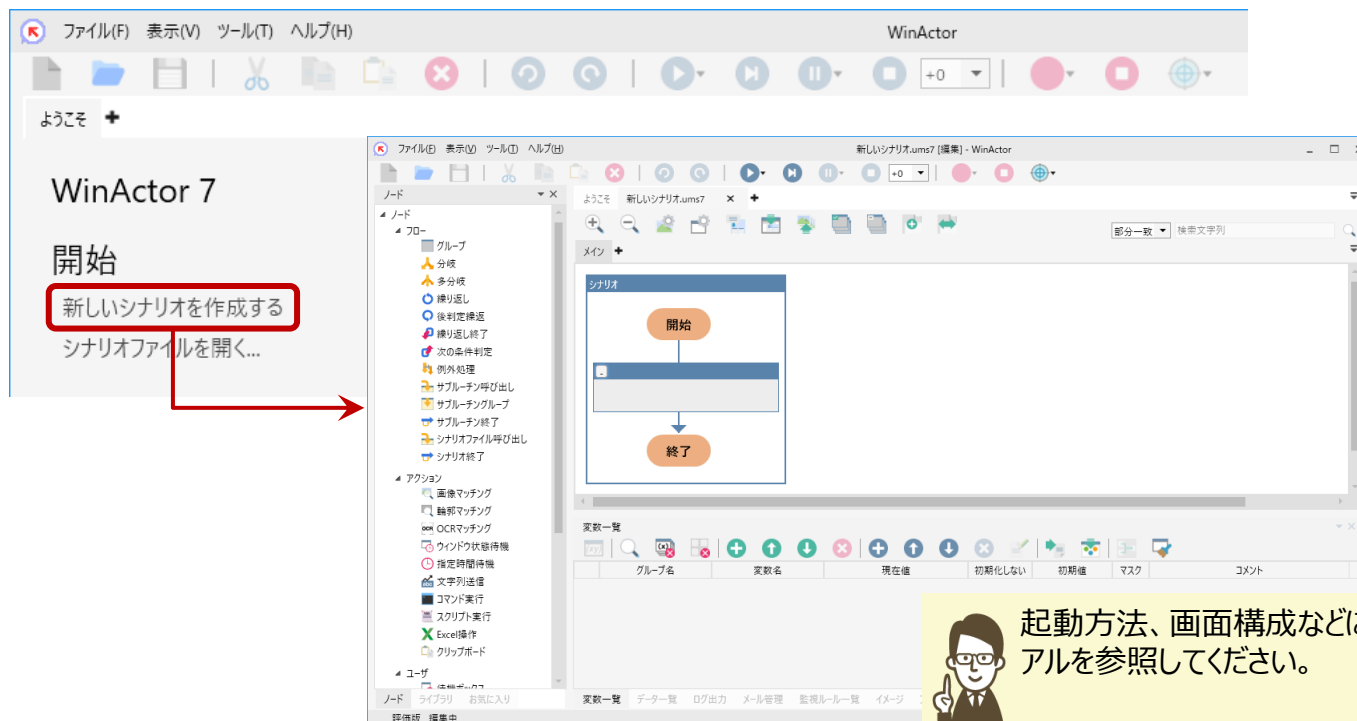
5

WebページからExcel台帳への転記を自動化する

準備

WinActorの起動

- 1 スタートメニューなどからWinActorを起動します。
- 2 「新しいシナリオを作成する」をクリックすると、WinActorのメイン画面が表示されます。



起動方法、画面構成などについては操作マニュアルを参照してください。

5

WebページからExcel台帳への転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」

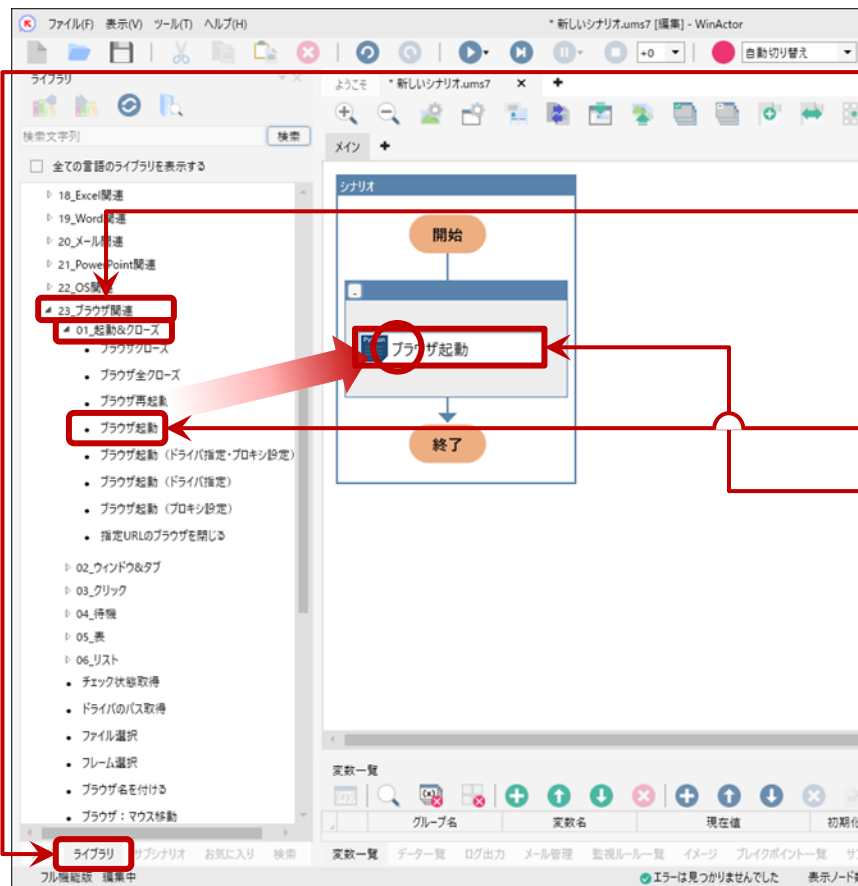


5-1 Webページの起動とログイン

5

WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン



1 <ライブラリ> をクリックします。

2 <23_ブラウザ関連> をダブルクリックして、リストを展開し、さらに <01_起動&クローズ> をダブルクリックします。

3 <ブラウザ起動> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

5

WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン

プロパティ Python実行

名前 ブラウザ起動

コメント

設定 スクリプト 注釈 バージョン情報

[タイムアウト(ミリ秒)(任意)]：以降のシナリオ実行の中で、タイムアウト指定の入力欄が
無いライブラリのWebページの要素を検索する時間
です。

[タイムアウト設定]でノードで指定が選択された
場合、本項目の値が有効になります。
未設定の場合、10,000ミリ秒となります。

ブラウザ名 値⇒ edge01

ブラウザ種類 edge

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値⇒

更新 元に戻す

5 <ブラウザ名> に、「値⇒ edge01」を入力します。

6 <ブラウザ種類> のプルダウンリストから<edge>を選択します。

7 <更新> ボタンをクリックします。

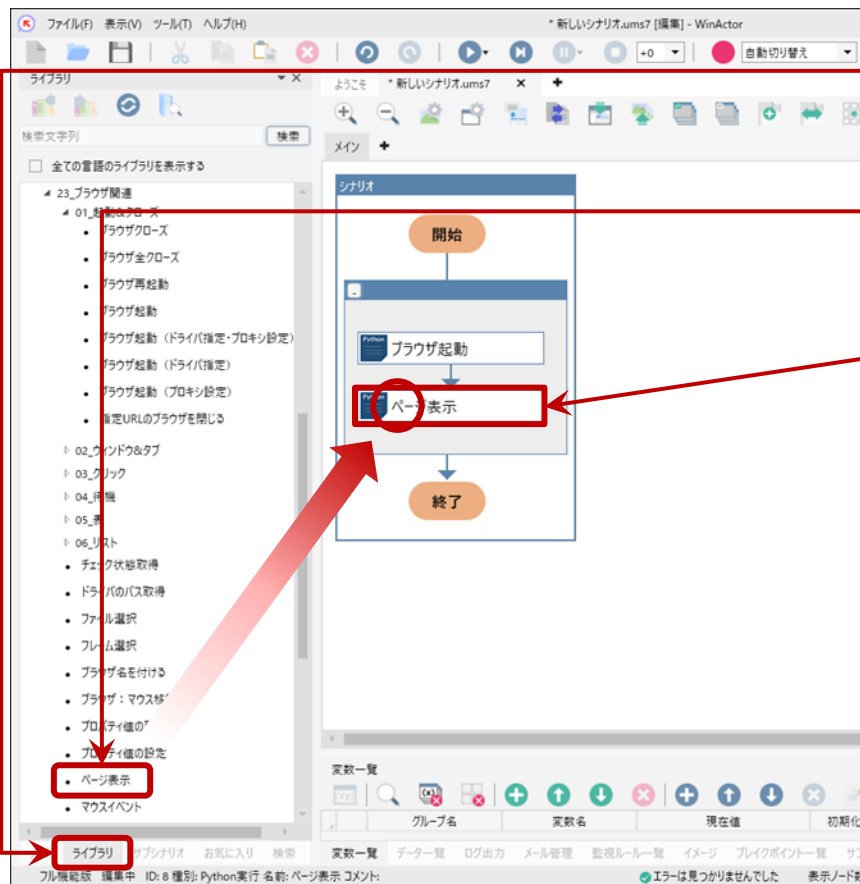


Chromeを使う人は<ブラウザ種類>のプルダウンリストから<Chrome>を選択してください。

5

WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン



8 <ライブラリ> をクリックします。

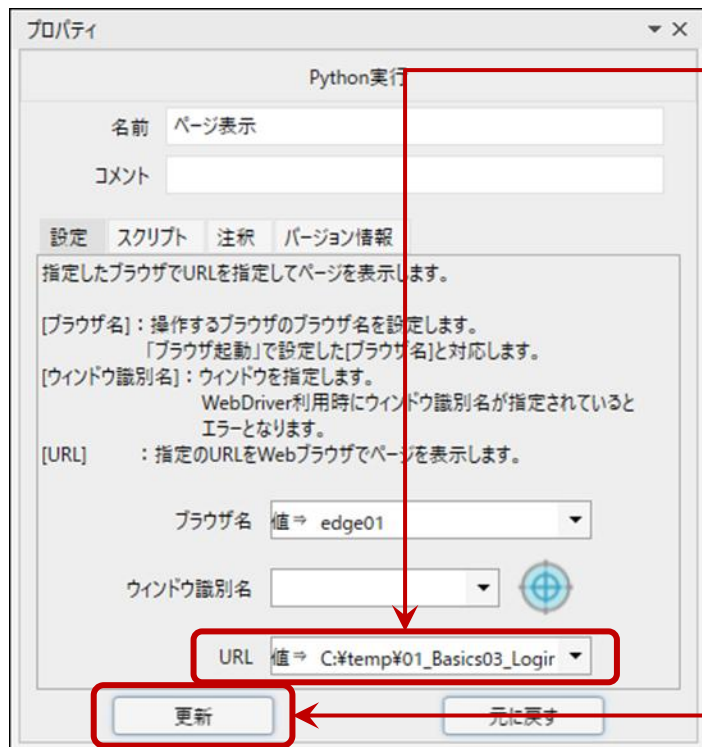
9 <ページ表示> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

10 配置したライブラリをダブルクリックします。(プロパティを表示)

5

WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン

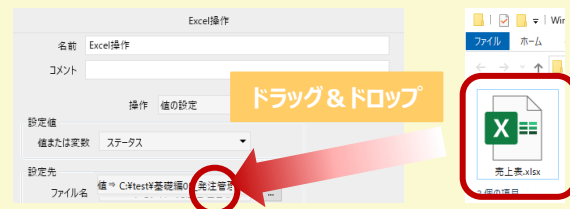


11 <URL> に 01_Basics03_Login.html をドラッグ & ドロップします。

12 <更新> ボタンをクリックします。

ファイルパスの設定は、直接手入力する方法以外にも上記のように対象ファイルをドラッグ & ドロップすることでも可能です。

画面に表示しているフォルダの中のファイルパスを、WinActorのノードやライブラリ設定したい時には、直接ドラッグ & ドロップする設定方法が便利です。選択するファイルの間違いを減らして、スムーズにシナリオ作成を進めましょう。



ドラッグ & ドロップ



5

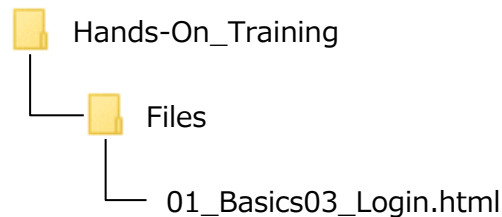
WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン

13 「01_Basics03_Login.html」を右クリックして、「プログラムから開く」から「Microsoft Edge」を選択します。



使用するファイル



デモシステムを起動するブラウザーには
Edge または Chrome を使用してください。

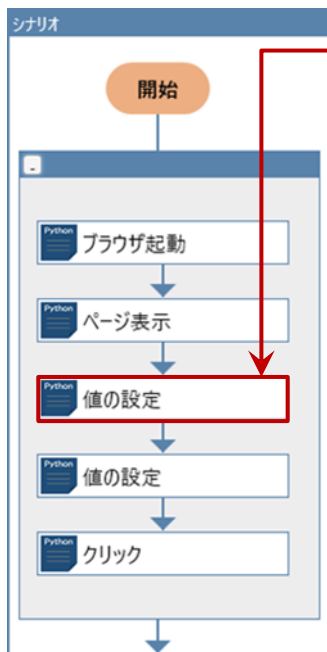


Chromeを使う人は「プログラムから開く」から
「Google Chrome」を選択してください。

5

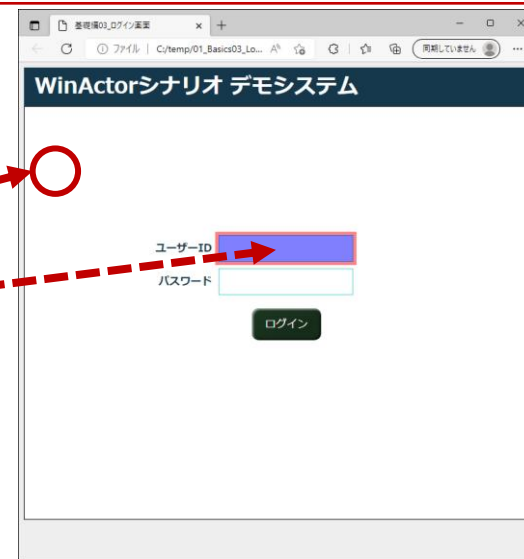
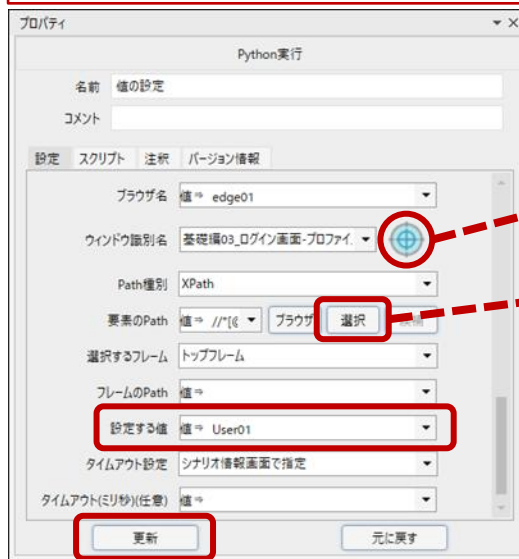
WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン



14

＜値の設定＞ ライブラリを配置して、操作対象となるウィンドウを設定します。
ユーザーIDに値が入るよう、＜要素のPath＞をユーザーIDのテキストボックスに指定します。



15

＜設定する値＞ に、「値⇒ User01」と入力し、＜更新＞ ボタンをクリックします。

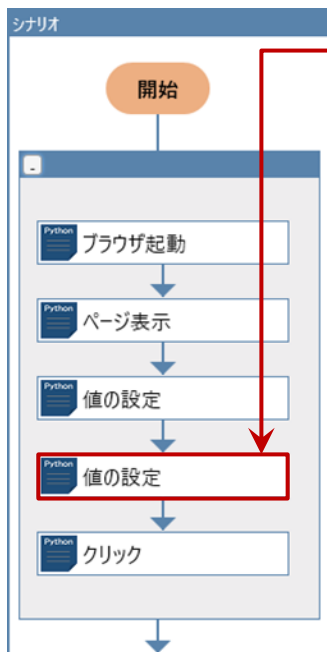


＜値の設定＞ については、
 “3 Webページのクリック&入力を自動化する”
 の3-6を参考に設定してください。

5

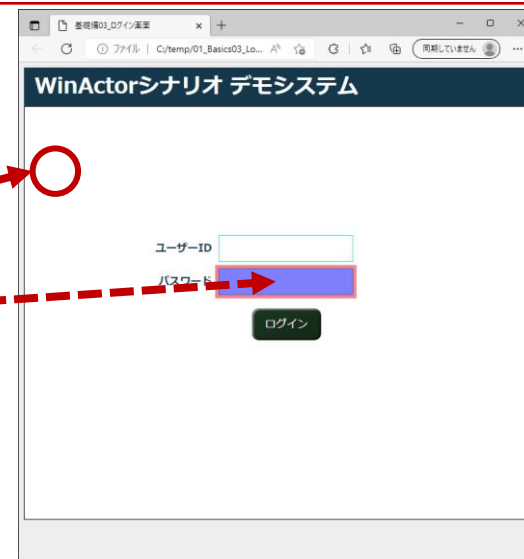
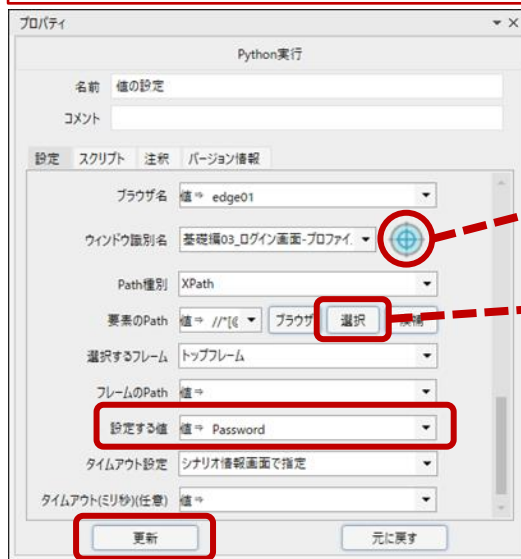
WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン



16

＜値の設定＞ ライブラリを配置して、操作対象となるウィンドウを設定します。
パスワードに値が入るよう、＜要素のPath＞をパスワードのテキストボックスに指定します。



17

＜設定する値＞に、「値⇒ Password」と入力し、＜更新＞ボタンをクリックします。

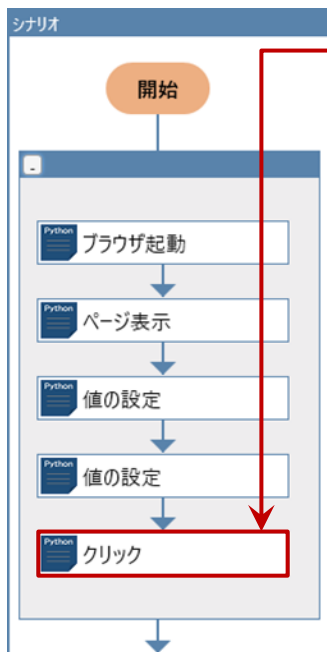


＜値の設定＞については、
”3 Webページのクリック&入力を自動化する”
の3-6を参考に設定してください。

5

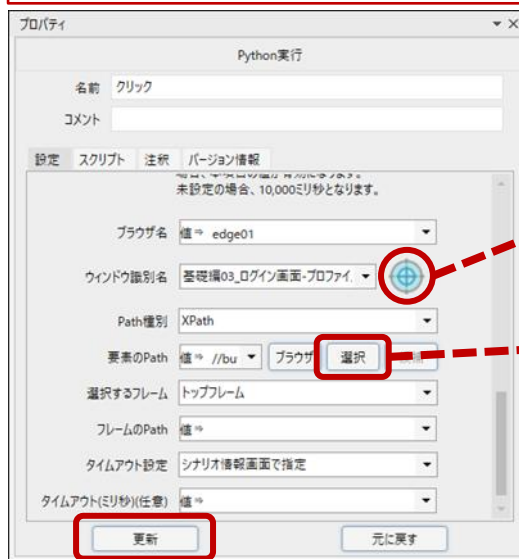
WebページからExcel台帳への転記を自動化する

5-1 Webページの起動とログイン



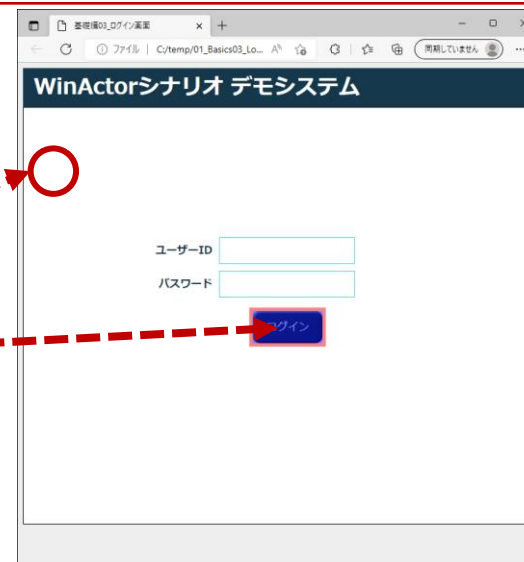
18

＜クリック＞ ライブラリを配置して、操作対象となるウィンドウを設定します。
ログインを実行するよう＜要素のPath＞に、ログインボタンを指定します。



19

＜更新＞ ボタンをクリックします。



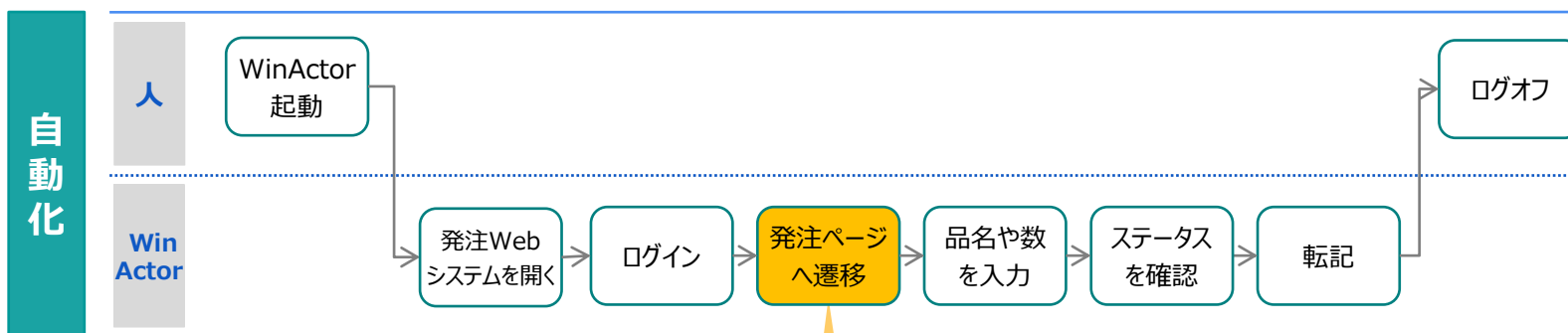
＜クリック＞ については、
”3 Webページのクリック&入力を自動化する”
の3-9を参考に設定してください。

5

WebページからExcel台帳への転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」



5-2

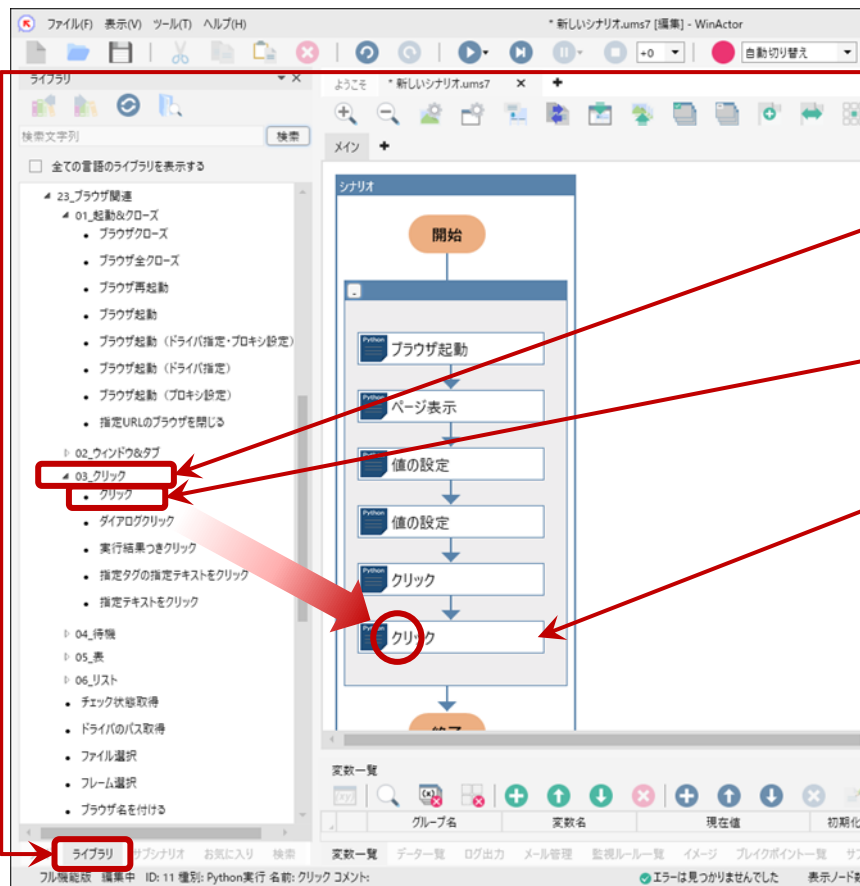
発注一覧への遷移

5

WebページからExcel台帳への転記を自動化する

5-2

発注一覧への遷移



1 <ライブラリ> をクリックします。

2 <03_クリック> をダブルクリックして、リストを展開します。

3 <クリック> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

5

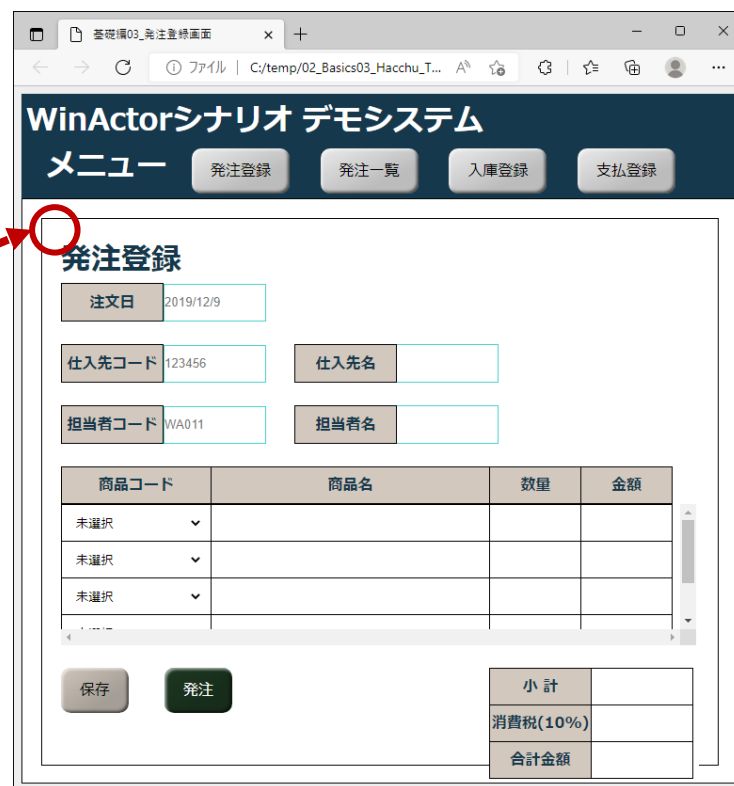
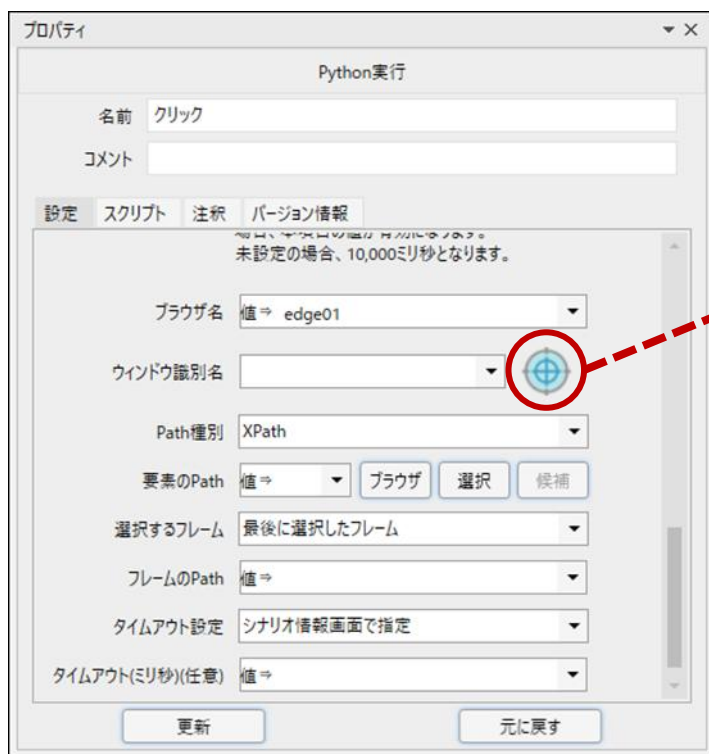
WebページからExcel台帳への転記を自動化する

5-2

発注一覧への遷移

5 <ウィンドウ識別名>の青いスコープボタンをクリックします。

6 操作したい画面上にマウスカーソルを移動し、クリックします。



5

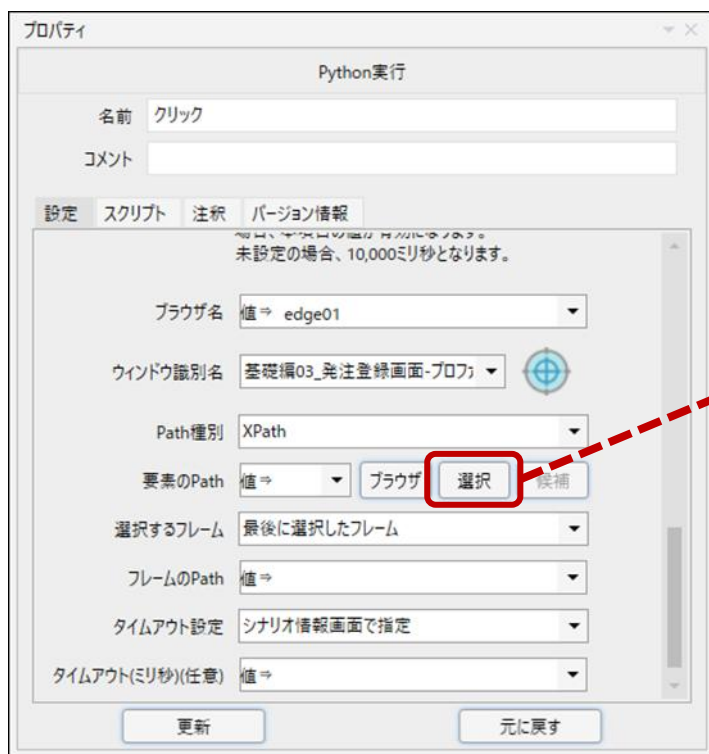
WebページからExcel台帳への転記を自動化する

5-2

発注一覧への遷移

7 <要素のPath> の<選択> ボタンをクリックします。

8 <発注一覧> ボタンにマウスカーソルを移動し、色が変わったら、クリックします。



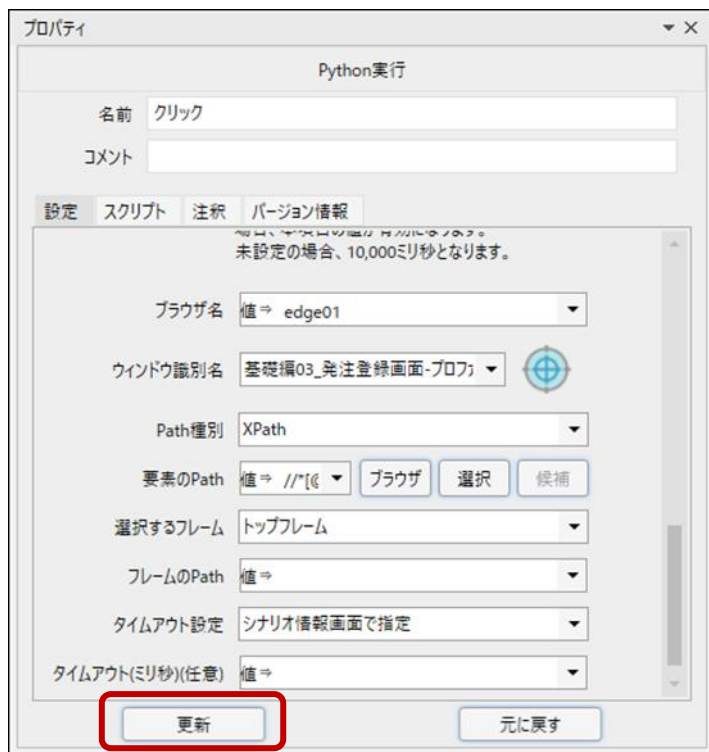
5

WebページからExcel台帳への転記を自動化する

5-2

発注一覧への遷移

9 <更新> ボタンをクリックします。

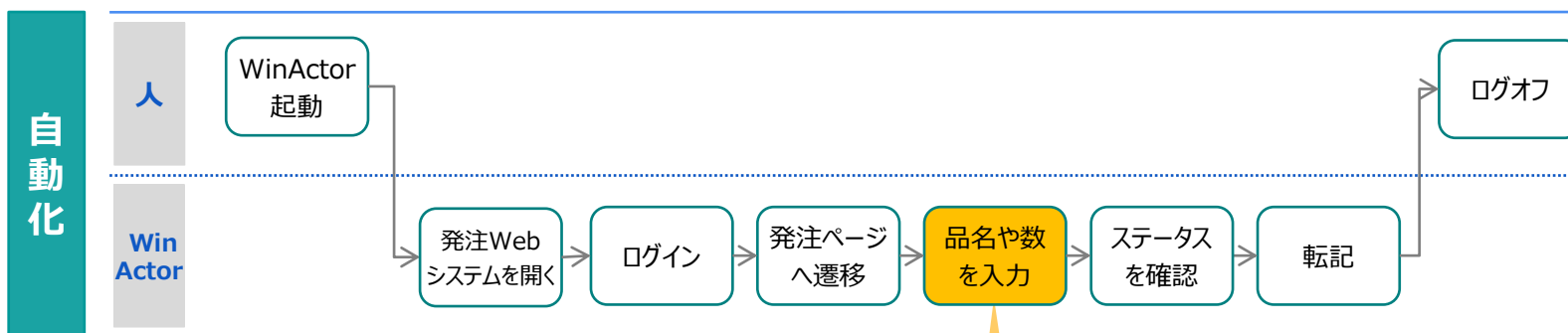


5

WebページからExcel台帳への転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」



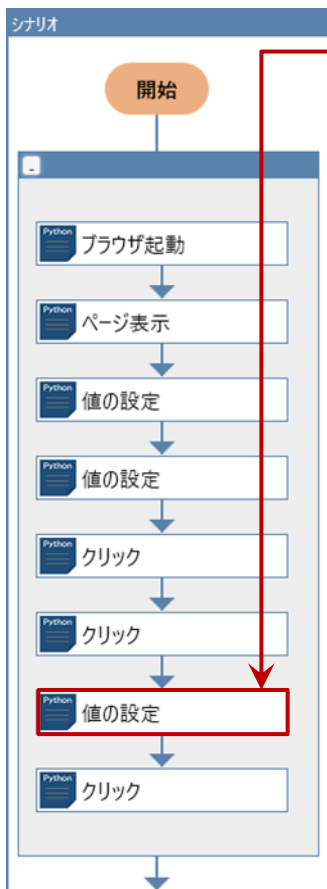
5-3

発注一覧での商品検索

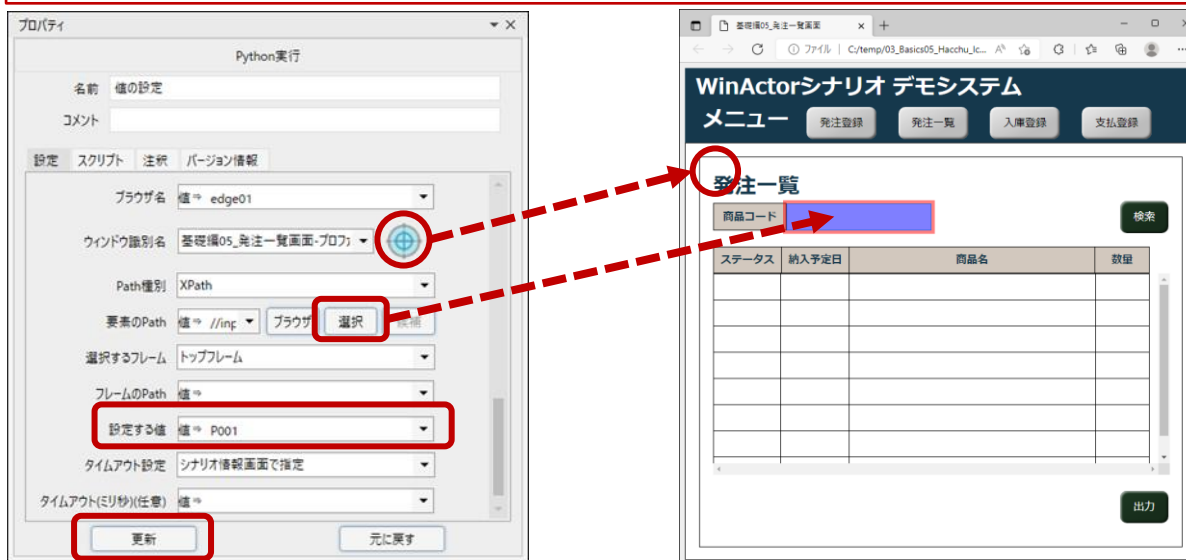
5

WebページからExcel台帳への転記を自動化する

5-3 発注一覧での商品検索



- 1 他のプロパティと同様に操作対象となるウィンドウを設定します。
商品コードに値が入るよう、＜要素のPath＞を商品コードのテキストボックスに指定します。



- 2 ＜設定する値＞に、「値⇒ P001」と入力し、＜更新＞ボタンをクリックします。



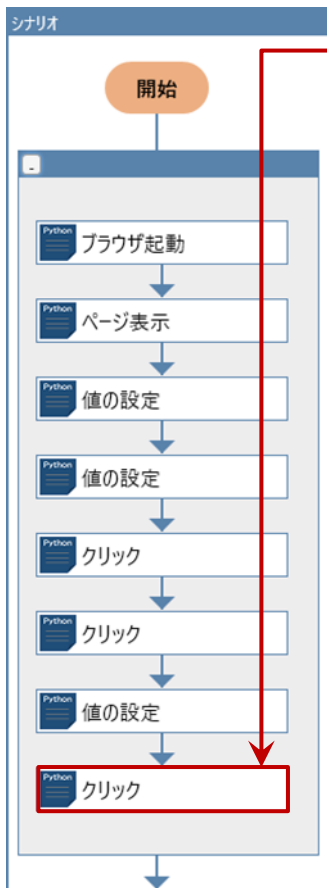
＜値の設定＞については、
“3 Webページのクリック&入力を自動化する”
の3-6を参考に設定してください。

5

WebページからExcel台帳への転記を自動化する

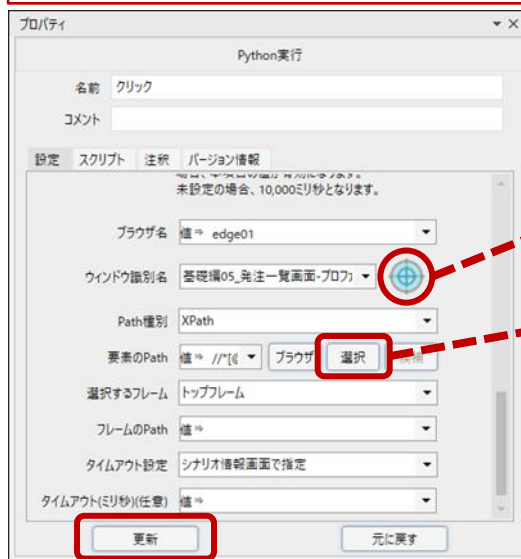
5-3

発注一覧での商品検索



3

他のプロパティと同様に操作対象となるウィンドウを設定します。
検索を実行するよう<要素のPath>に、検索ボタンを指定します。



4

<更新> ボタンをクリックします。



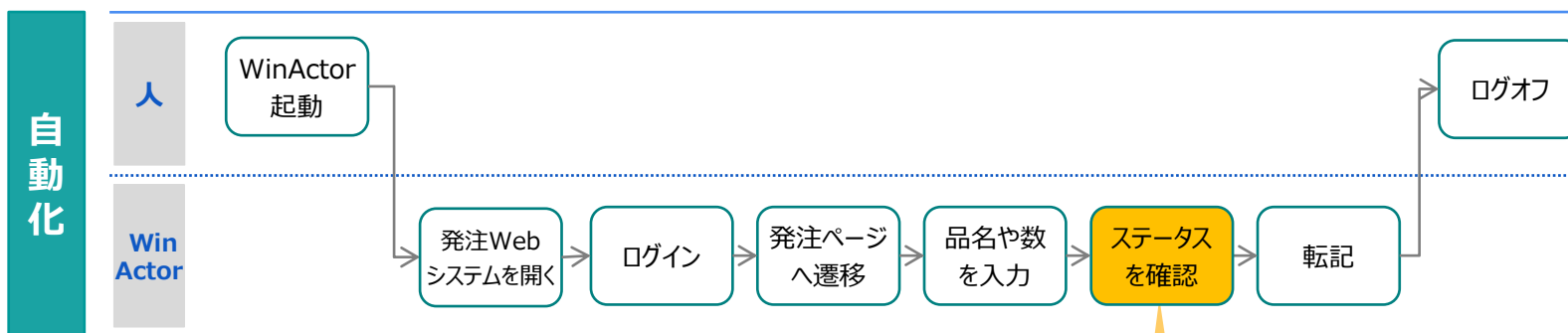
<クリック> については、
"3 Webページのクリック&入力を自動化する"
の3-9を参考に設定してください。

5

WebページからExcel台帳への転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」



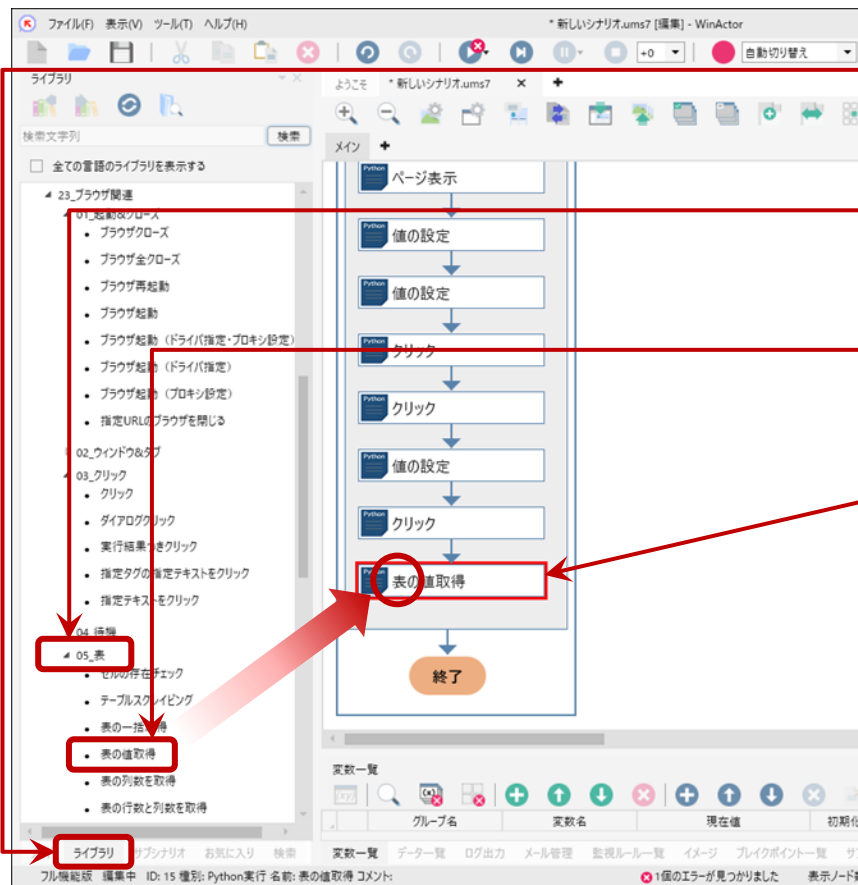
5-4

発注一覧のステータス取得

5

WebページからExcel台帳への転記を自動化する

5-4 発注一覧のステータス取得



1 <ライブラリ> をクリックします。

2 <05_表> をダブルクリックして、リストを展開します。

3 <表の値取得> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

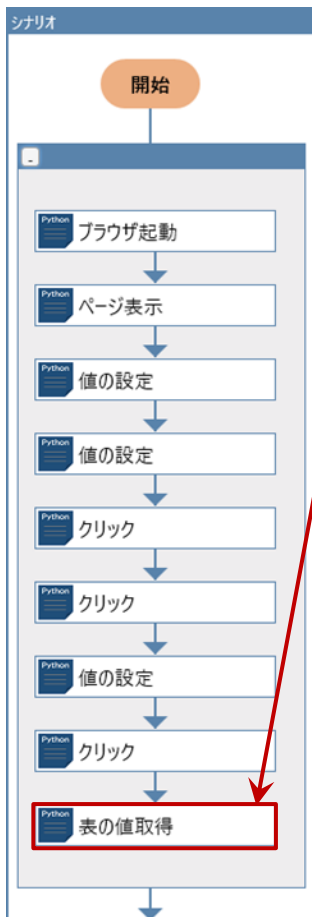
4 配置したライブラリをダブルクリックします。(プロパティを表示)

5

WebページからExcel台帳への転記を自動化する

5-4

発注一覧のステータス取得



5

他のプロパティと同様に操作対象となるウィンドウを設定します。
 <表のPath> は、画面上の表部分を指定します。

プロパティ Python実行

名前 表の値取得

コメント

設定 スクリプト 注釈 バージョン情報

ブラウザ名 値⇒ edge01

ウィンドウ識別名 基礎編05_発注一覧画面-プロパティ

Path種別 XPath

表のPath 値⇒ //*[6] ブラウザ **選択**

選択するフレーム トップフレーム

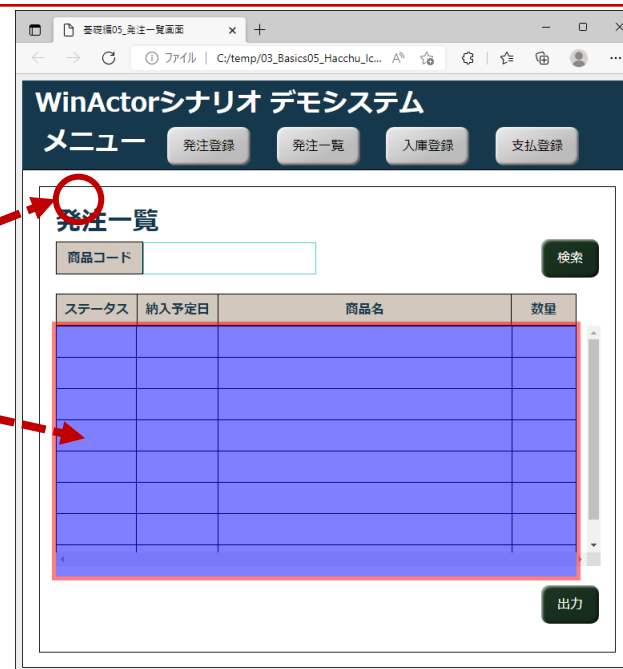
フレームのPath 値⇒

行番号 値⇒

列番号 値⇒

値を格納する変数 変数名を選択

更新 元に戻す

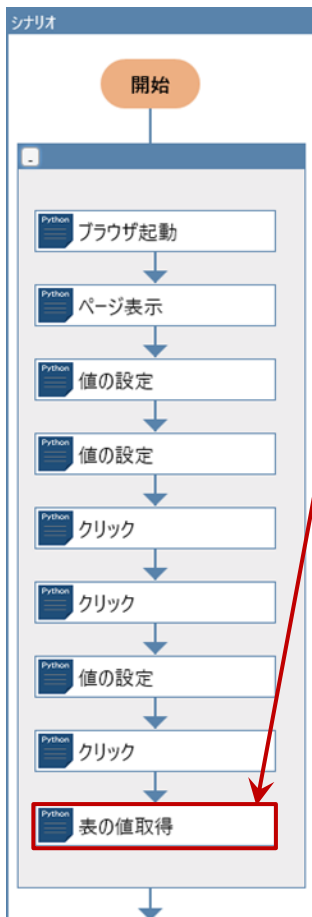


5

WebページからExcel台帳への転記を自動化する

5-4

発注一覧のステータス取得



6 <表の値取得>の残りのプロパティは以下のように設定します。

プロパティ Python実行

名前 表の値取得

コメント

設定 スクリプト 注釈 バージョン情報

ブラウザ名 値⇒ edge01

ウィンドウ識別名 基礎編05_発注一覧画面-プロパ

Path種別 XPath

表のPath 値⇒ //*[6] ブラウザ 選択 候補

選択するフレーム トップフレーム

フレームのPath 値⇒

行番号 値⇒ 1

列番号 値⇒ 1

値を格納する変数 ステータス

更新 元に戻す

6.1 <行番号>と<列番号>の両方に「値⇒ 1」を入力します。

6.2 <値を格納する変数>に「ステータス」と入力します。

6.3 <更新> ボタンをクリックします。
変数登録確認のダイアログは<はい>をクリックします。

WinActor

変数"ステータス"は変数一覧に存在しません。
変数を登録しますか?

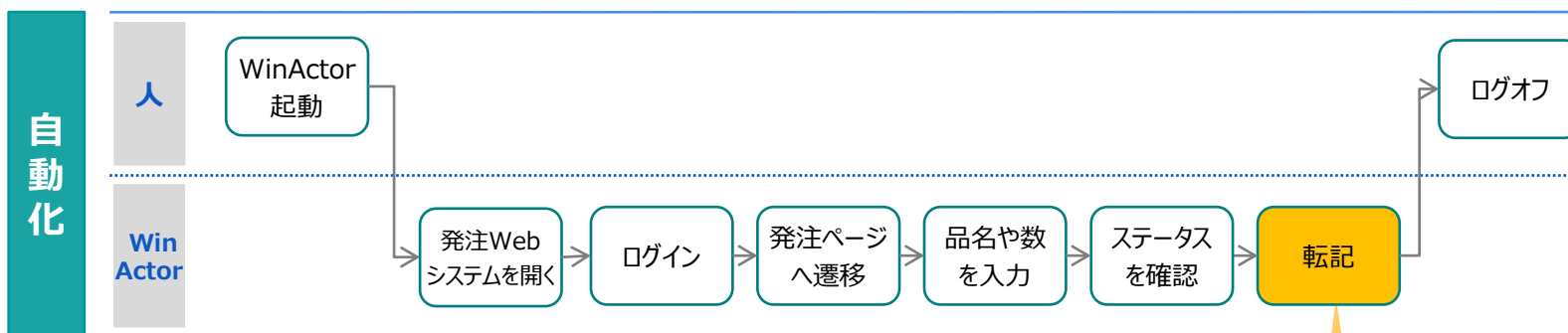
はい いいえ

5

WebページからExcel台帳への転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」



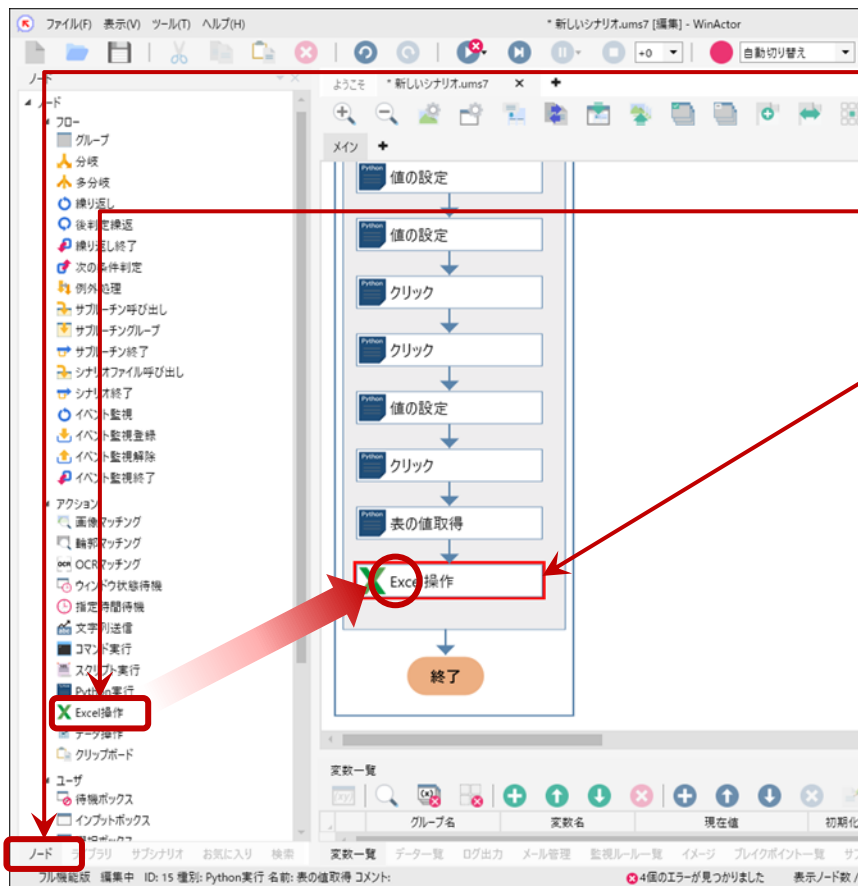
5-5

Excelへのステータス入力

5

WebページからExcel台帳への転記を自動化する

5-5 Excelへのステータス入力



1 <ノード> をクリックします。

2 <Excel操作> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したライブラリをダブルクリックします。(プロパティを表示)

5

WebページからExcel台帳への転記を自動化する

5-5

Excelへのステータス入力

Excel操作

名前 Excel操作

コメント

操作 値の設定

設定値

値または変数 ステータス

設定先

ファイル名 値⇒ C:\temp\04_Basics05_Hacchu_

シート名 値⇒ Sheet1

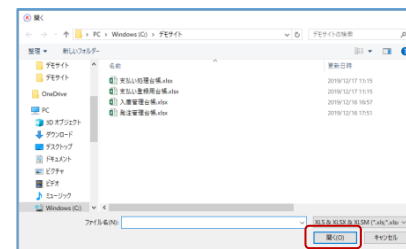
セル位置 値⇒ J5

更新 元に戻す

4 <操作> をクリックし、<値の設定> を選択します。

5 <値または変数> をクリックし、<ステータス> を選択します。

6 <ファイル名> 参照ボタンをクリックすると、エクスプローラーが開きます。参照先となる「04_Basics05_Hacchu_Kanri_Daicho.xlsx」を指定し、開きます。



7 <シート名> に、「値⇒ Sheet1」を入力し（参照ボタンも利用できます）
<セル位置> に、「値⇒ J5」を指定します。

8 <更新> ボタンをクリックします。

5

WebページからExcel台帳への転記を自動化する

5-6 シナリオ全体の確認

1 作成したシナリオが以下の通りになっていることを確認します。



シナリオの実行と保存は、
 “3 Webページのクリック&入力を自動化する”
 の3-11、3-12と同様に行います。

5-7 シナリオの実行

5-8 シナリオの保存

Tips

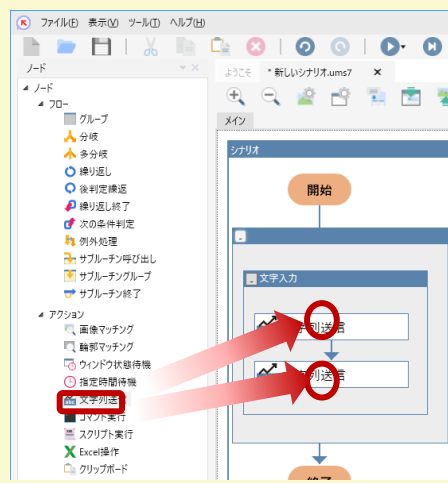
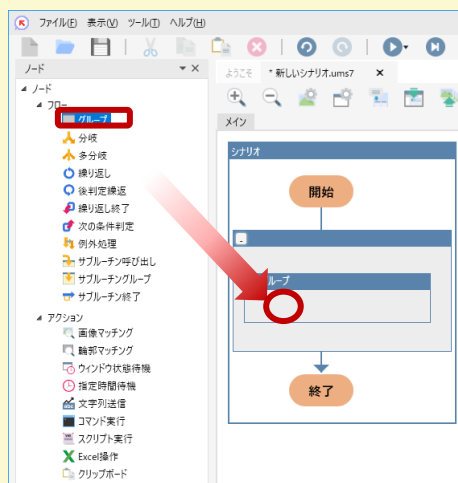
グループ(ノード)の活用

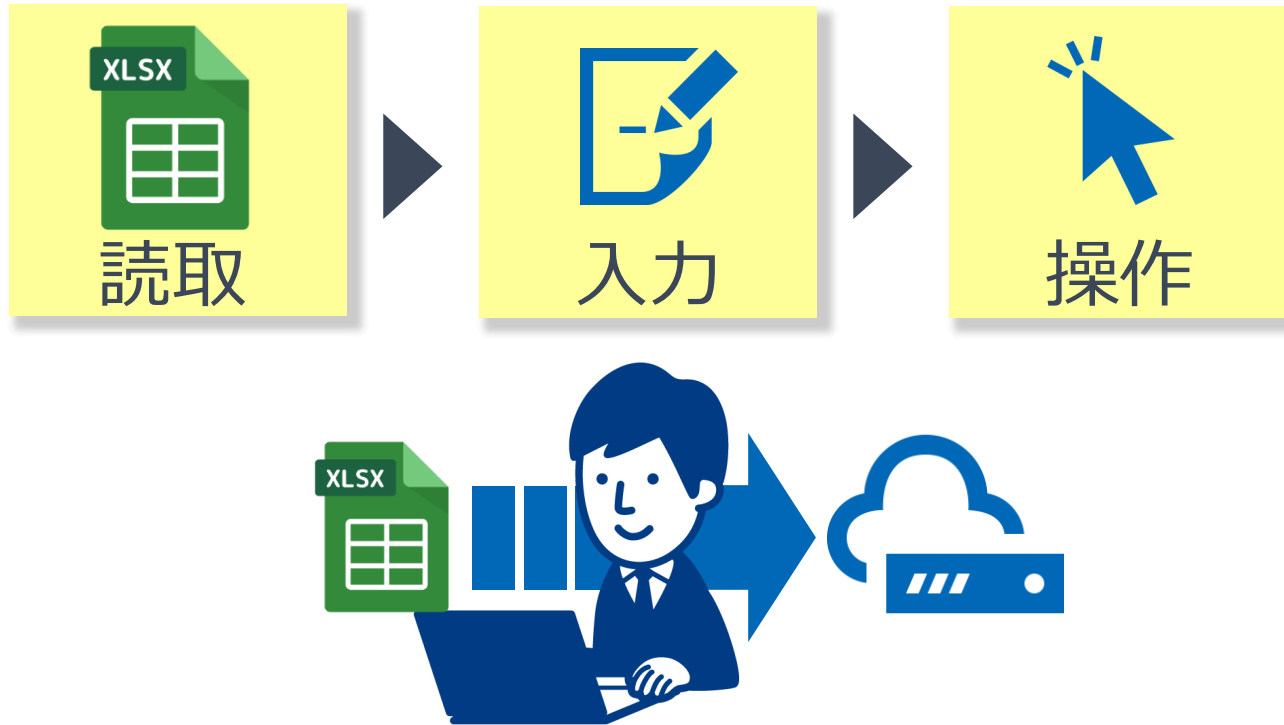


グループ(ノード)の活用することでシナリオが見やすくなります。

シナリオに似たようなノードやライブラリが並ぶ場合は、「グループ」のノードを利用するとシナリオの組み立てが見やすく、分かりやすくなります。

「グループ」の中にノードやライブラリを置くことができるため、ステップの区切りや画面の区切りなどでグループ化することで、シナリオ作成作業の段取りをつけたり、共同で開発している他のユーザーから見ても分かりやすいシナリオとなります。





Excel台帳から社内システムへの転記を自動化する

導入

宇院さんは発注ステータスを台帳に転記する作業も自動化でき、ひと安心しました。
今度は自社の社内システムに入庫情報を転記する作業も自動化しようとしています。



「今まではWebシステムとExcelを使った自動化をしてきたけれども、自社の社内システムへ転記する作業も同じようにできそうだな。」



「もちろんです、WinActorはブラウザーやExcelなどのアプリケーション以外にも、社内システムでの利用も可能です。今回も既存の部品を組み合わせで自動化をしていきましょう。」



「自社の社内システムは使いづらいので、自動化できると心理的にも助かります！」



「今回も手作業とWinActorを使った自動化とで業務がどう変わるか確認しましょう。」

6

シナリオ作成 ～基礎編.3～



「WinActorを使った自動化では以下のように業務が変わります。」

手作業

人

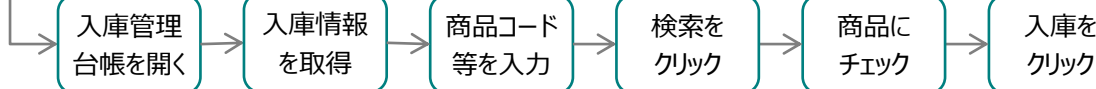


自動化

人



WinActor



ログオフ



「基本的には、前回と同じような流れになるのですね。」



「今回も、＜Excel操作＞というノードを利用します。」



「前回使いましたよね？今回はどのような使い方をするんですか？」

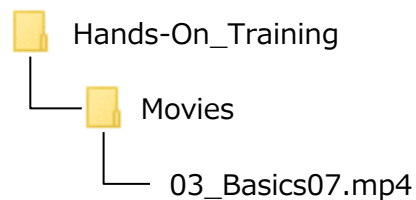
「今回は、Excelファイルを開き、値を取得し、社内システムへ転記するという使い方をします！」



「前回Excel操作ノードは、書き込み以外に読み込みにも使用できるとお伝えしましたが、実際の業務では部内で共有している台帳から、社内システムに情報を転記することが多く発生します。ここでしっかりと設定方法と取得した値の使用方法を学んでおきましょう！」



「それでは実際のWinActorの動きを見てみましょう。以下フォルダのファイル名をダブルクリックして再生してください。」



「それでは実際にシナリオを一緒に作っていきましょう！」

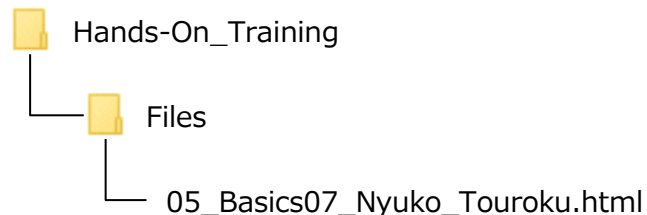
Excel台帳から社内システムへの転記を自動化する

準備 社内システム画面の表示

- 1 「05_Basics07_Nyuko_Touroku.html」を右クリックして、「プログラムから開く」から「Microsoft Edge」を選択します。

ステータス	商品名	数量
☐		
☐		
☐		
☐		
☐		
☐		

使用するファイル



デモシステムを起動するブラウザには
Edge または Chrome を使用してください。



Chromeを使う人は「プログラムから開く」から
「Google Chrome」を選択してください。

Excel台帳から社内システムへの転記を自動化する

準備

入庫管理台帳の確認

自動保存 入庫管理台帳.xlsx - Excel

ファイル ホーム 挿入 ページレイアウト 数式 データ 校閲 表示 ヘルプ 実行したい作業を入力し

J20

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										

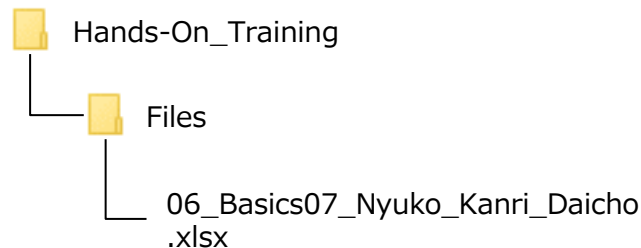
入庫管理台帳

通番	入庫転記日	倉庫コード	依頼部署	依頼者氏名	商品コード	備考
1	2019/12/25	0001	経理部	鈴木太郎	P001	
2	2019/12/25	0002	経理部	田村恵一	P001	
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						

Sheet1

準備完了

確認するファイル



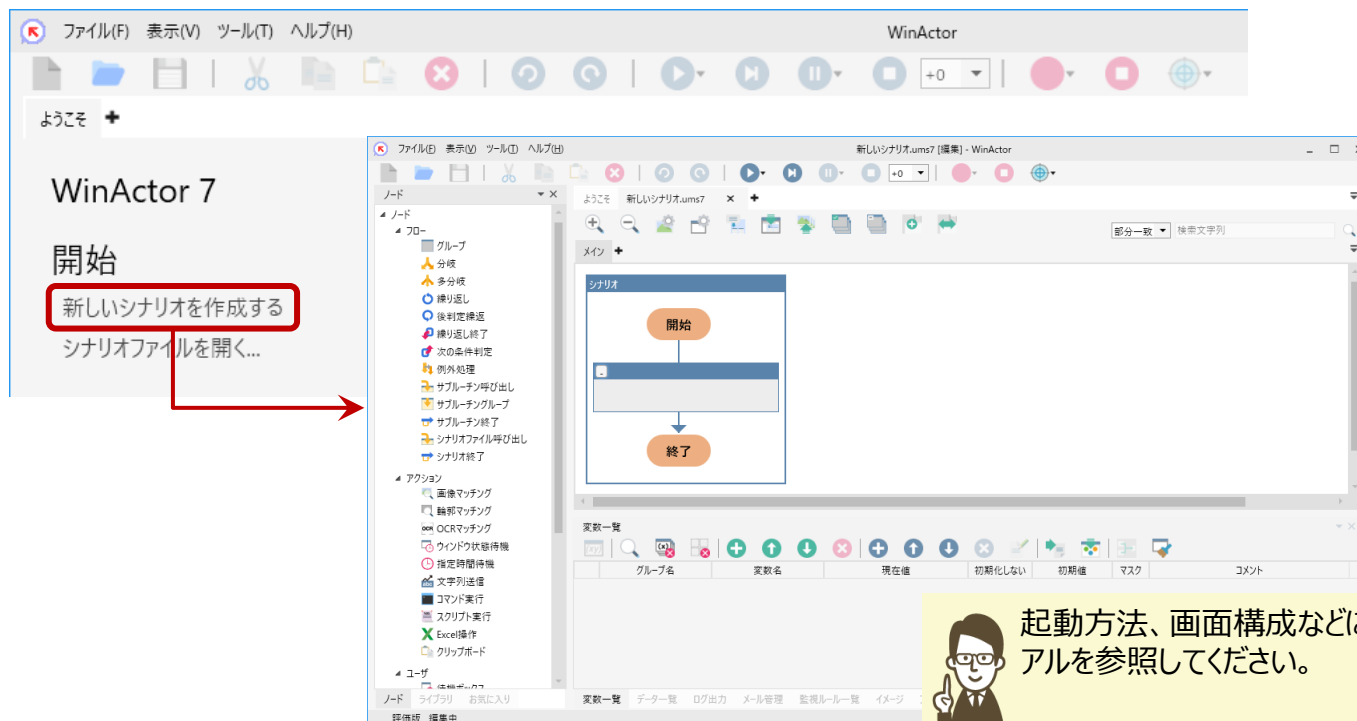
転記元となる入庫管理台帳を確認します。
業務で使っているため台帳は見慣れているかもしれませんが、シナリオ作成では普段あまり意識していないシート・セルの位置（行と列）などが重要になってきます。



Excel台帳から社内システムへの転記を自動化する

準備 WinActorの起動

- 1 スタートメニューなどからWinActorを起動します。
- 2 「新しいシナリオを作成する」をクリックすると、WinActorのメイン画面が表示されます。



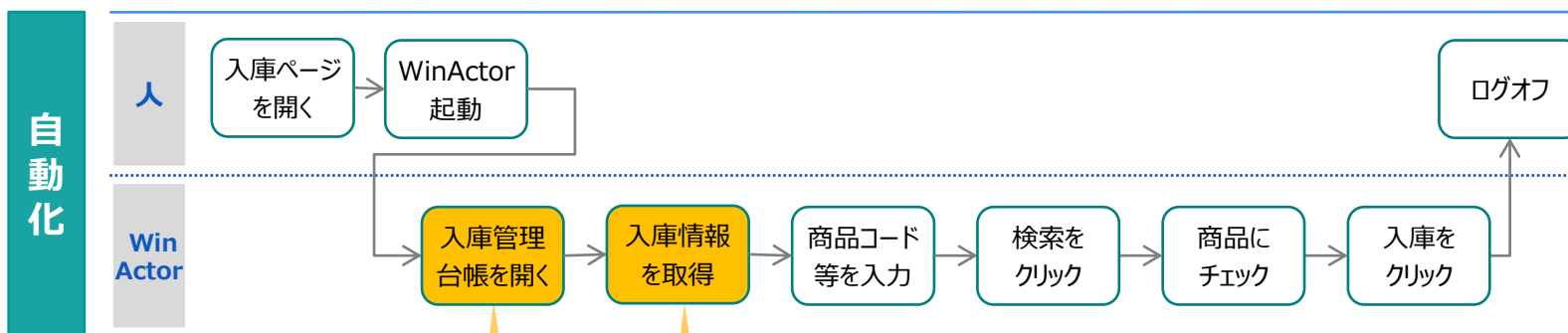
起動方法、画面構成などについては操作マニュアルを参照してください。

7

Excel台帳から社内システムへの転記を自動化する

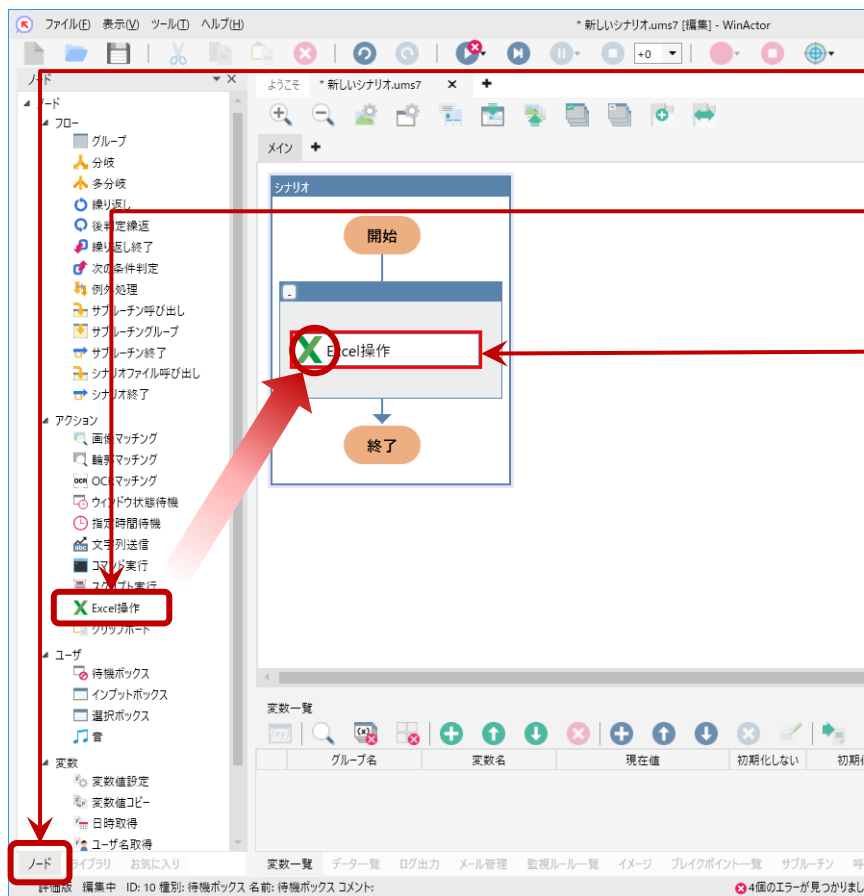


「これから以下の業務についてシナリオを作成していきます。」



7-1 Excelからの値読み込み（ノード利用）

7-1 Excelからの値読み込み（ノード利用）



1 <ノード> をクリックします。

2 <Excel操作> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したノードをダブルクリックします。(プロパティを表示)

7-1 Excelからの値読み込み（ノード利用）

Excel操作

名前 Excel操作

コメント

操作 値の取得

取得元

ファイル名 値⇒ C:\temp\06_Basics07_Nyuko_K

シート名 値⇒ Sheet1

セル位置 値⇒ C5

格納先

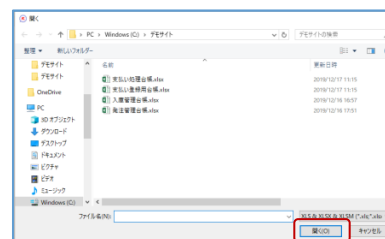
変数 入庫転記日

更新

元に戻す

4 <操作> のプルダウンリストから<値の取得> を選択します。

5 <ファイル名> 参照ボタンをクリックすると、エクスプローラーが開きます。参照先となる「06_Basics07_Nyuko_Kanri_Daicho.xlsx」を指定し、開きます。



6 <シート名> に、「値⇒ Sheet1」を入力し（参照ボタンも利用できます）
<セル位置> に、「値⇒ C5」を指定します。

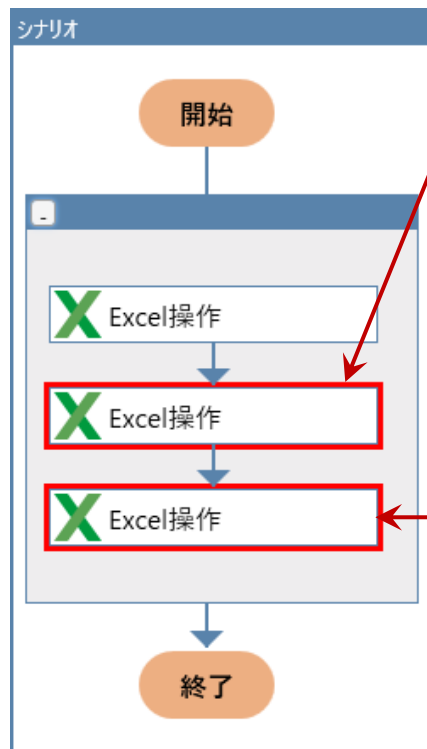
7 <変数> をクリックし、「入庫転記日」と入力します。

8 <更新> ボタンをクリックします。変数登録確認のダイアログは<はい> をクリックします。

7

Excel台帳から社内システムへの転記を自動化する

7-1 Excelからの値読み込み（ノード利用）



9 入庫管理台帳から倉庫コードの値を読み込むため、＜Excel操作＞ノードを配置し以下のように設定します。（＜セル位置＞に「値⇒ D5」を、＜変数＞に「倉庫コード」を入力）

Excel操作

名前 Excel操作

コメント

操作 値の取得

取得元

ファイル名 値⇒ C:\temp\06_Basics07_Nyuko_K ...

シート名 値⇒ Sheet1 ...

セル位置 値⇒ D5

格納先

変数 倉庫コード

10 入庫管理台帳から商品コードの値を読み込むため、＜Excel操作＞ノードを配置し以下のように設定します。（＜セル位置＞に「値⇒ G5」を、＜変数＞に「商品コード」を入力）

Excel操作

名前 Excel操作

コメント

操作 値の取得

取得元

ファイル名 値⇒ C:\temp\06_Basics07_Nyuko_K ...

シート名 値⇒ Sheet1 ...

セル位置 値⇒ G5

格納先

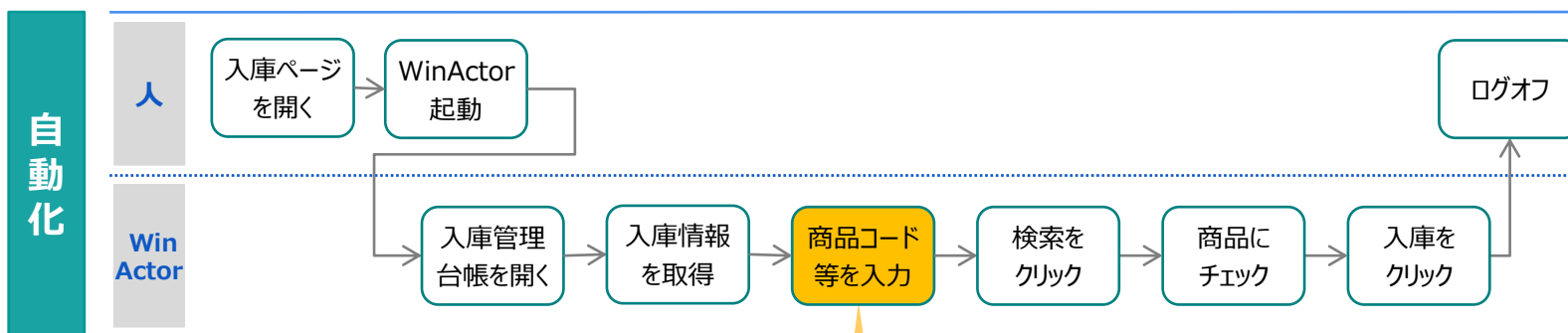
変数 商品コード

7

Excel台帳から社内システムへの転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」



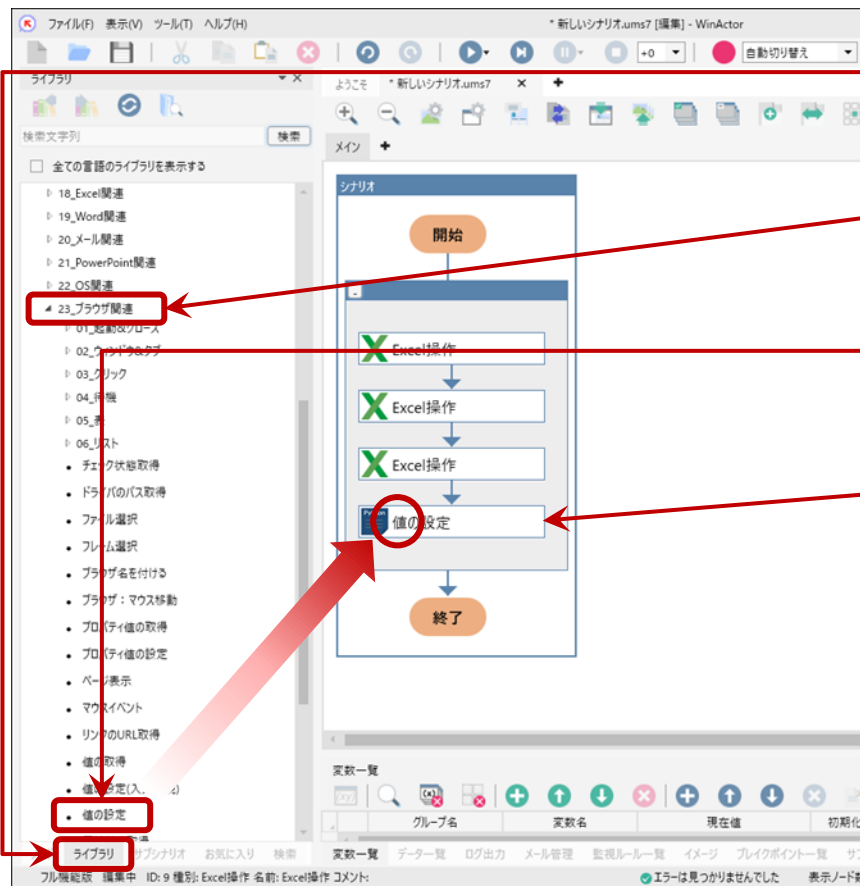
7-2

入庫登録画面への入力

7

Excel台帳から社内システムへの転記を自動化する

7-2 入庫登録画面への入力



1 <ライブラリ> をクリックします。

2 <23_ブラウザ関連> をダブルクリックして、リストを展開します。

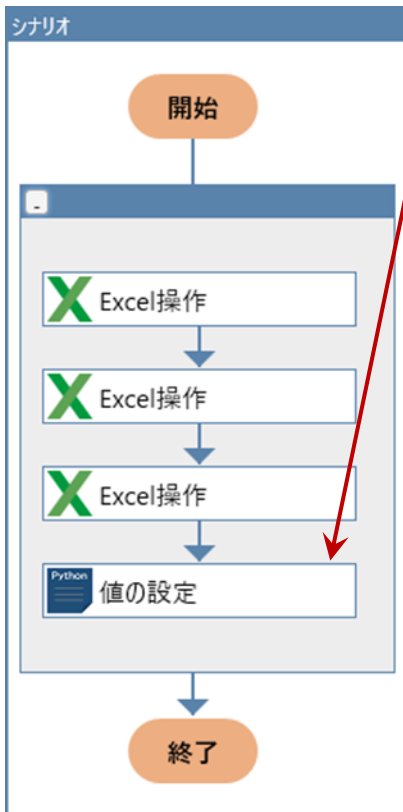
3 <値の設定> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

7

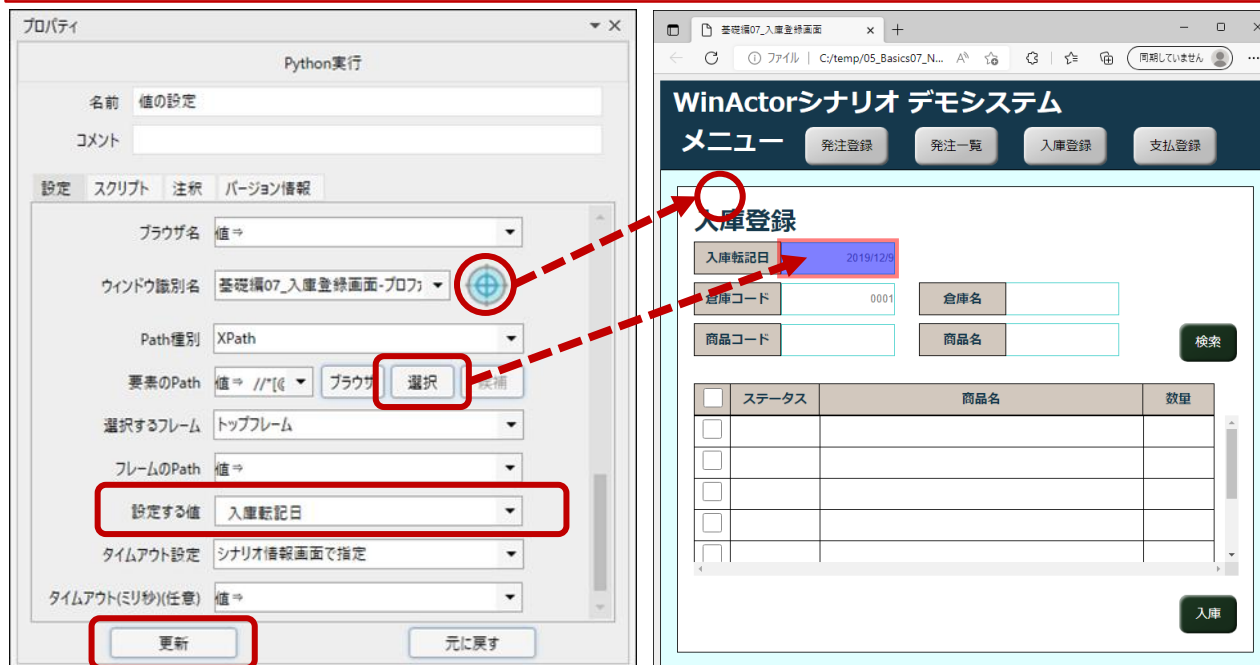
Excel台帳から社内システムへの転記を自動化する

7-2 入庫登録画面への入力



5

操作対象となるウィンドウを設定します。
 <要素のPath> は、<入庫転記日> のテキストボックスを指定します。



6

<設定する値> は「入庫転記日」をプルダウンリストから選択し、<更新> ボタンをクリックします。

7

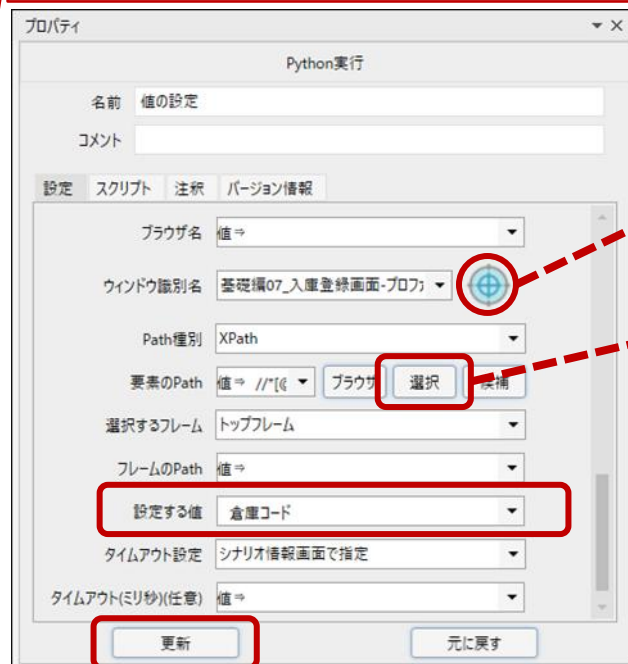
Excel台帳から社内システムへの転記を自動化する

7-2 入庫登録画面への入力



7

操作対象となるウィンドウを設定します。
 <要素のPath> は、<倉庫コード> のテキストボックスを指定します。



8

<設定する値> は「倉庫コード」をプルダウンリストから選択し、<更新> ボタンをクリックします。

7

Excel台帳から社内システムへの転記を自動化する

7-2

入庫登録画面への入力

シナリオ

開始

Excel操作

Excel操作

Excel操作

Python
値の設定Python
値の設定Python
値の設定

9

操作対象となるウィンドウを設定します。

<要素のPath> は、<商品コード> のテキストボックスを指定します。

プロパティ Python実行

名前 値の設定

コメント

設定 スクリプト 注釈 バージョン情報

ブラウザ名 値

ウィンドウ識別名 基礎帳07_入庫登録画面-プロパ

Path種別 XPath

要素のPath 値 //*[4] ブラウザ 選択

選択するフレーム トップフレーム

フレームのPath 値

設定する値 商品コード

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値

更新 元に戻す

WinActorシナリオ デモシステム

メニュー 発注登録 発注一覧 入庫登録 支払登録

入庫登録

入庫転記日 2019/12/9

倉庫コード 0001 倉庫名

商品コード 商品名

検索

ステータス	商品名	数量
☐		
☐		
☐		
☐		
☐		
☐		

入庫

10

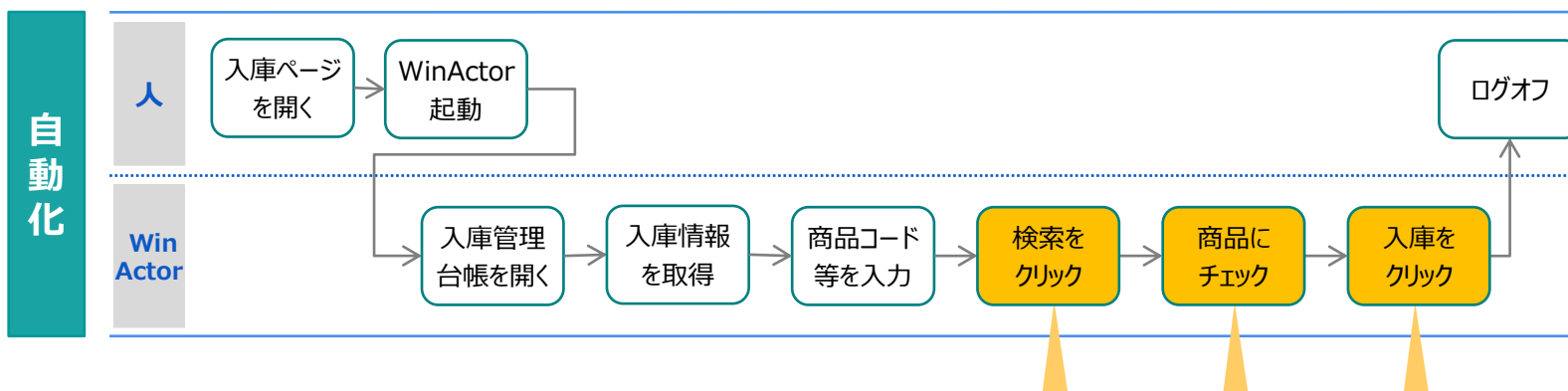
<設定する値> は「商品コード」をプルダウンリストから選択し、<更新> ボタンをクリックします。

7

Excel台帳から社内システムへの転記を自動化する



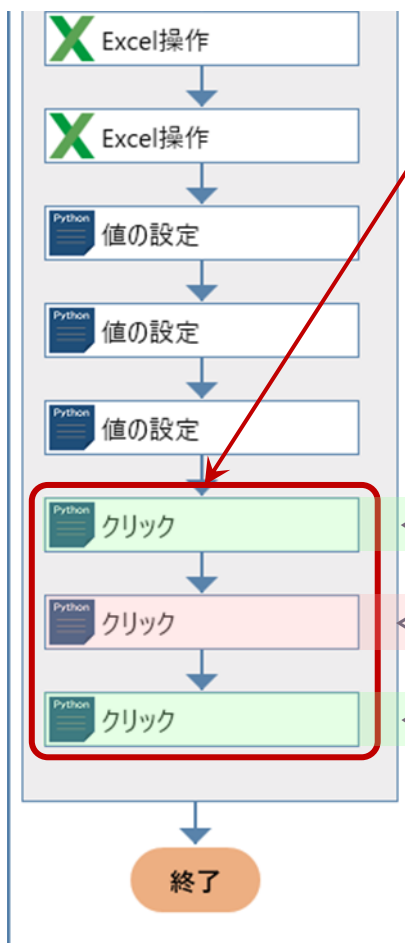
「これから以下の業務についてシナリオを作成していきます。」



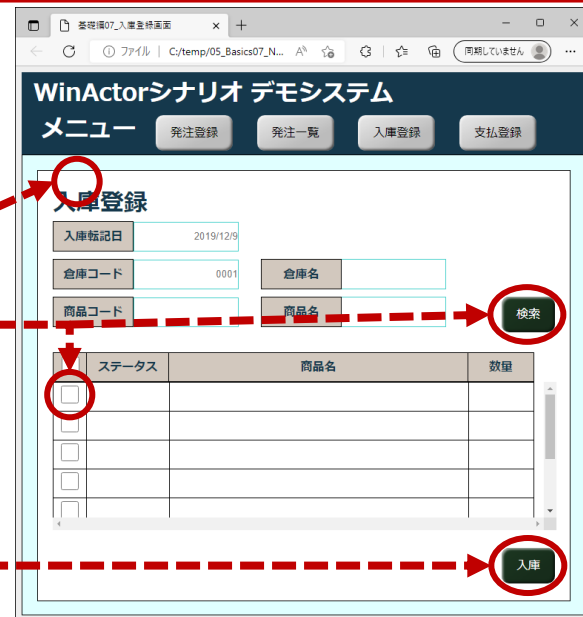
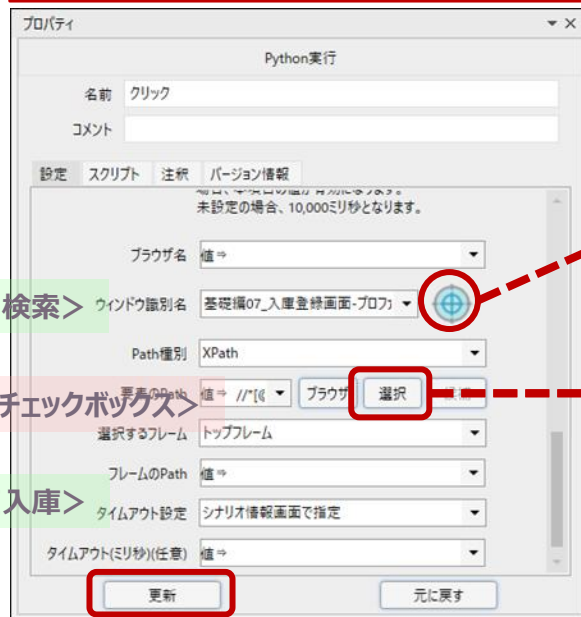
7-3

入庫登録画面での商品検索・入庫登録

7-3 入庫登録画面での商品検索・入庫登録



1 検索実行、明細チェック、入庫実行をするために<クリック> ライブラリを3つ配置します。
 <ウィンドウ識別名>には、基礎編07_入庫登録画面を指定し、
 <要素のPath>は、<検索ボタン> <チェックボックス> <入庫ボタン>をライブラリ1つに
 対して、1つずつ指定します。



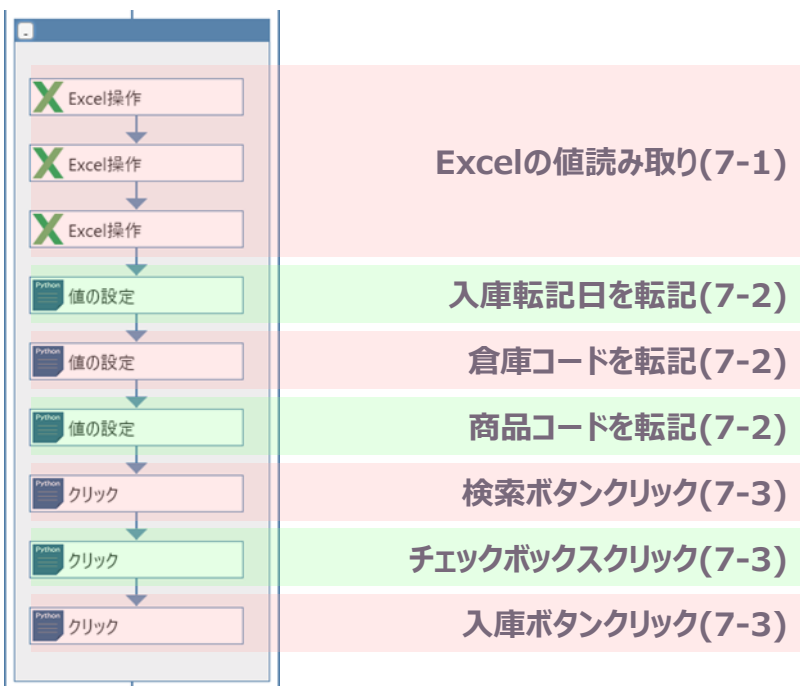
2 <更新> ボタンをクリックします。

7

Excel台帳から社内システムへの転記を自動化する

7-4 シナリオ全体の確認

1 作成したシナリオが以下の通りになっていることを確認します。



シナリオの実行と保存は、
"3 Webページのクリック&入力を自動化する"
の3-11、3-12と同様に行います。

7-5 シナリオの実行

7-6 シナリオの保存



Excel台帳から別のExcel台帳への
転記を自動化する

導入

宇院さんはすっかりWinActorが使えるようになってきました。
その噂を聞いて経理部の方から頼まれごとをされたようです。



「いくつもある台帳から必要な数字だけをもってきて、支払処理台帳を作してほしいと頼まれてしまった。ただ、今までExcelから転記とExcelへの転記をしてきたからできそうな気がするな。」



「できたという経験の積み重ねが大事です！その調子ですね。
ExcelやOutlookなどのOfficeアプリケーションの自動化もWinActorの強みのひとつです。
今回も既存の部品を組み合わせで自動化をしていきましょう。」



「Officeアプリケーションに強いのは助かります。業務で使用的是ExcelやWordにOutlookなので。」



「今回も手作業とWinActorを使った自動化とで業務がどう変わるか確認しましょう。」



「WinActorを使った自動化では以下のように業務が変わります。」

手作業

人

発注管理
台帳を開くセルを
コピーする支払処理
台帳を開くセルに
ペーストする

自動化

人

WinActor
起動Win
Actor発注管理
台帳を開くセルを
コピーする支払処理
台帳を開くセルに
ペーストする

「これなら今までの経験でシナリオが作れそうです！」



「今回は、＜Excel操作（値の設定 2、値の取得 2）＞というライブラリを利用します。」



「Excel操作ノードの使い方は既に習いましたよね？」

「以前学んだExcel操作ノードはセルを指定するのに行列一緒での指定しかできませんでしたよね？ Excel操作（値の設定 2、値の取得 2）は列と行を別々に指定できるのです！」



繰り返し処理

書き込み

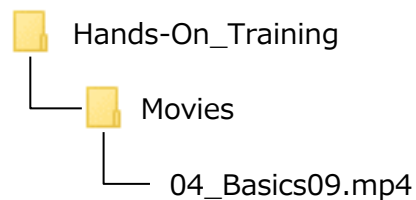


読み込み

「実践編で繰り返し処理を学ば際に、列と行を別々に指定できることが大きなメリットになります。今回は繰り返し処理は行いませんが使って慣れておきましょう！」



「それでは実際のWinActorの動きを見てみましょう。以下フォルダのファイル名をダブルクリックして再生してください。」



「それでは実際にシナリオを一緒に作っていきましょう！」

9

Excel台帳から別のExcel台帳への転記を自動化する

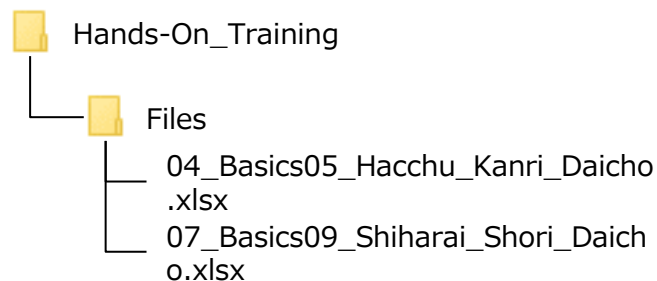
準備

発注管理台帳・支払処理台帳の確認

発注管理台帳										
通番	注文日	承認日	依頼部署	依頼者氏名	仕入れ先コード	担当者コード	商品コード	発注後ステータス	備考	
1	2019/12/25	2019/12/25	経理部	鈴木太郎	8811	WA011	P001			
2	2019/12/25	2019/12/25	経理部	田村恵一	8811	WA011	P002			
3	2019/12/26	2019/12/27	経理部	坂口素子	8811	WA011	P003			
4	2019/12/27	2019/12/27	経理部	横井保平	8811	WA011	P004			
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										

支払処理台帳										
通番	支払日	銀行コード	支払先コード	勘定コード	支払金額	依頼者氏名	商品コード			
1	2019年11月30日	NAT0001	987654	100990	8765000		P001			
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										

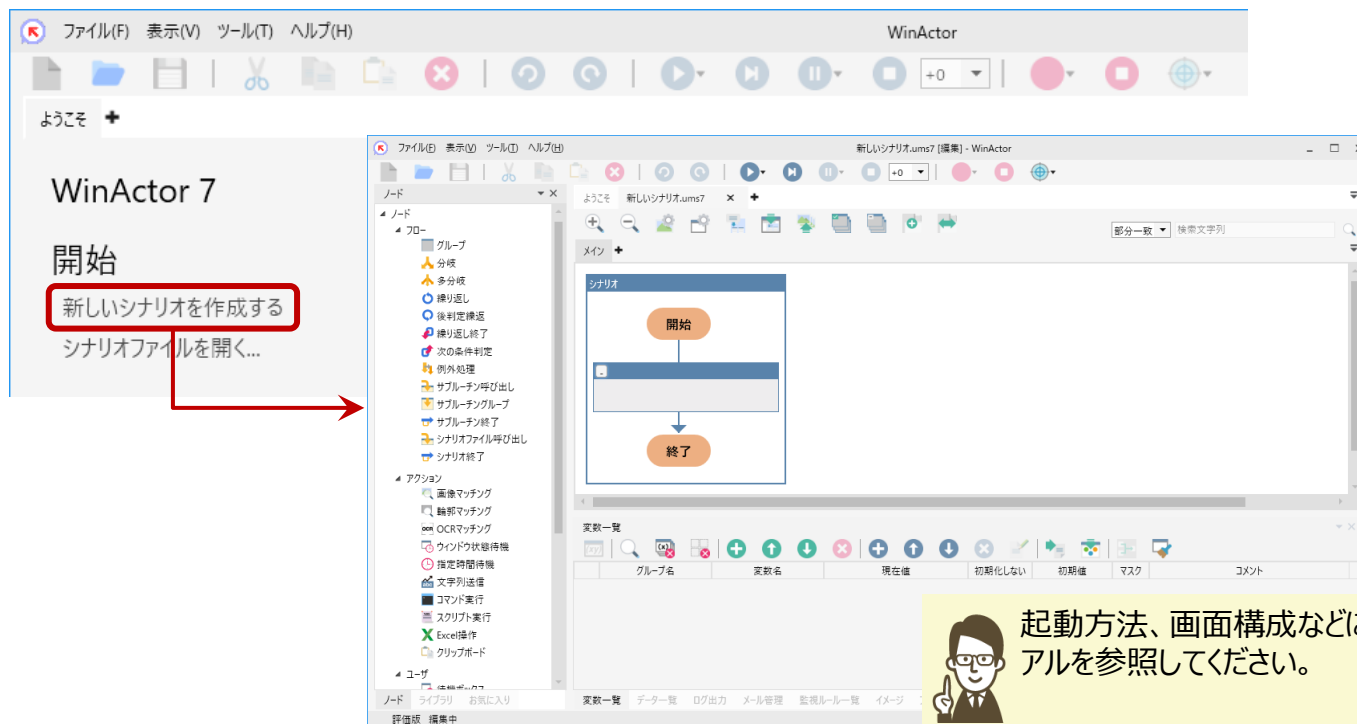
確認するファイル



今回は発注管理台帳から支払い処理台帳に転記を行います。

準備 WinActorの起動

- 1 スタートメニューなどからWinActorを起動します。
- 2 「新しいシナリオを作成する」をクリックすると、WinActorのメイン画面が表示されます。



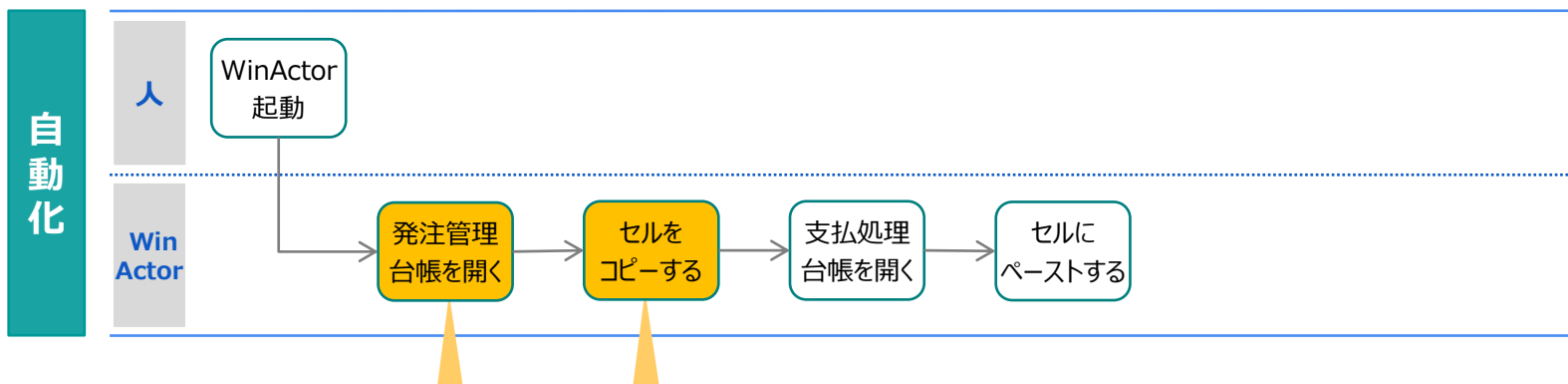
起動方法、画面構成などについては操作マニュアルを参照してください。

9

Excel台帳から別のExcel台帳への転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」

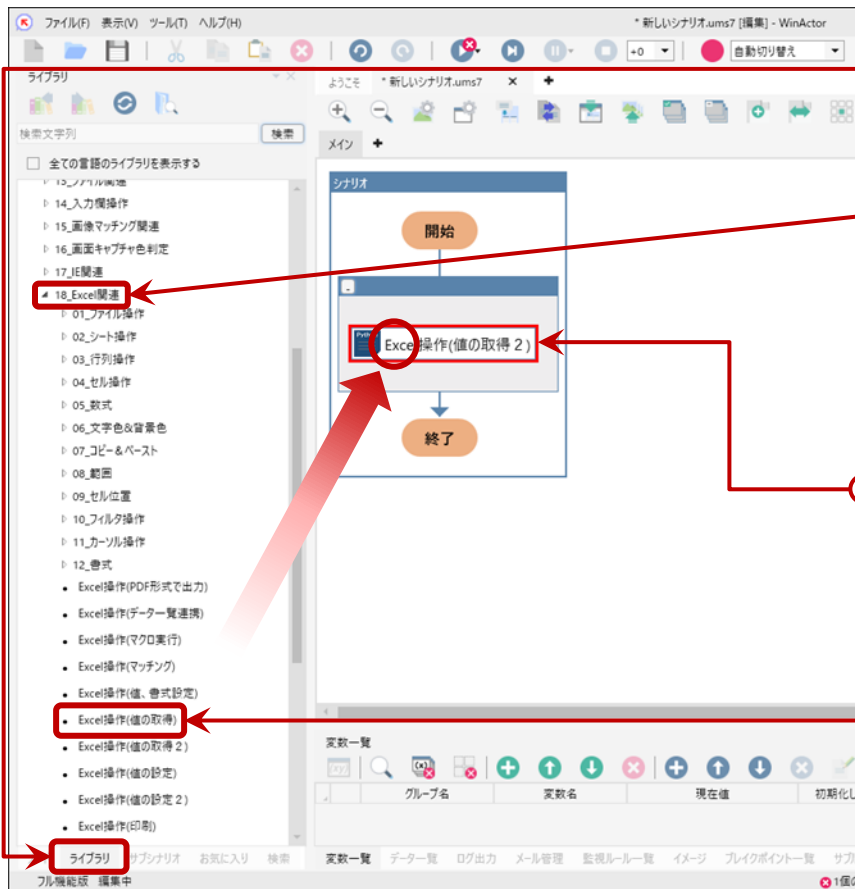


9-1 Excelからの値読み込み（ライブラリ利用）

9

Excel台帳から別のExcel台帳への転記を自動化する

9-1 Excelからの値読み込み（ライブラリ利用）



1 <ライブラリ> をクリックします。

2 <18_Excel関連> をダブルクリックして、リストを展開します。

3 <Excel操作(値の取得 2)> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

9

Excel台帳から別のExcel台帳への転記を自動化する

9-1 Excelからの値読み込み（ライブラリ利用）

プロパティ

Python実行

名前 Excel操作(値の取得2)

コメント

設定 スクリプト 注釈 バージョン情報

[格納先変数]
取得した値を格納する変数を指定してください。

ファイル名 値⇒ C:\temp\04_Basics05_Hacchu_Kanri_Daich

シート名 値⇒ Sheet1

セル(行) 値⇒ 5

セル(列) 値⇒ F

取得方法 value

格納先変数 依頼者氏名

更新 元に戻す

5

<ファイル名> に04_Basics05_Hacchu_Kanri_Daicho.xlsx
をドラッグ＆ドロップし、
<シート名> に「値⇒ Sheet1」
<セル(行)> に「値⇒ 5」
<セル(列)> に「値⇒ F」
<取得方法> に「value」
<格納先変数> に、依頼者氏名
を入力します。

6

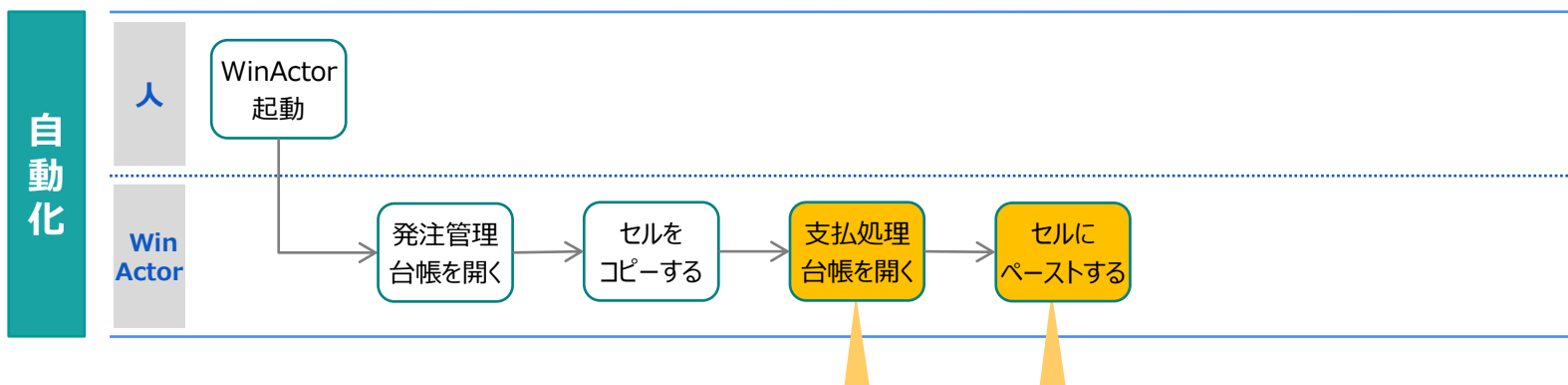
<更新> ボタンをクリックします。
変数登録確認のダイアログは <はい> をクリックします。

9

Excel台帳から別のExcel台帳への転記を自動化する



「これから以下の業務についてシナリオを作成していきます。」



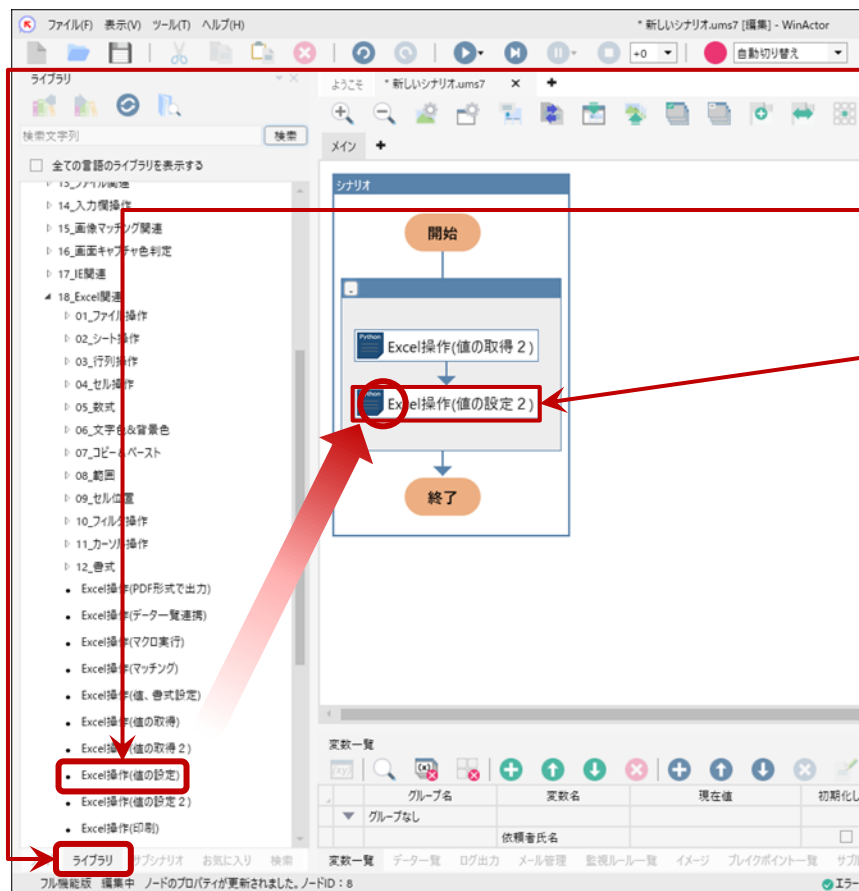
9-2

Excelへの値入力（ライブラリ利用）

9

Excel台帳から別のExcel台帳への転記を自動化する

9-2 Excelへの値入力（ライブラリ利用）



1 <ライブラリ> をクリックします。

2 <Excel操作(値の設定2)> をドラッグし、シナリオ編集エリア上でドロップします。

3 配置したライブラリをダブルクリックします。(プロパティを表示)

9

Excel台帳から別のExcel台帳への転記を自動化する

9-2 Excelへの値入力（ライブラリ利用）



4

<設定値> に「依頼者氏名」を入力し
<ファイル名> に
07_Basics09_Shiharai_Shori_Daicho.xlsx をドラッグ & ド
ロップ、
<シート名> に「値⇒ Sheet1」
<セル(行)> に「値⇒ 5」
<セル(列)> に「値⇒ H」
を入力します。

5

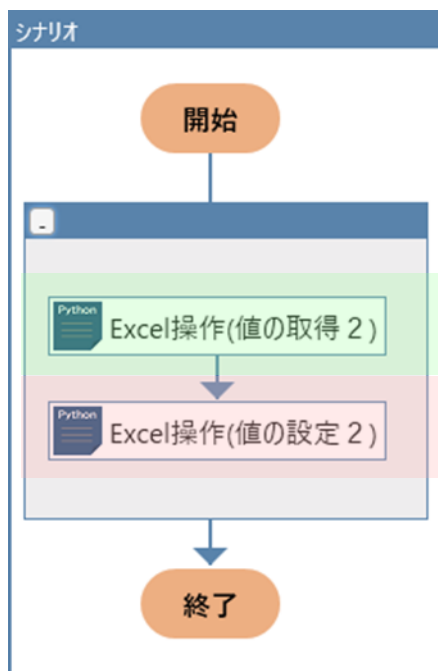
<更新> ボタンをクリックします。

9

Excel台帳から別のExcel台帳への転記を自動化する

9-3 シナリオ全体の確認

- 1 作成したシナリオが以下の通りになっていることを確認します。



発注管理台帳から依頼者を取得(9-1)

取得した情報を支払処理台帳に転記(9-2)

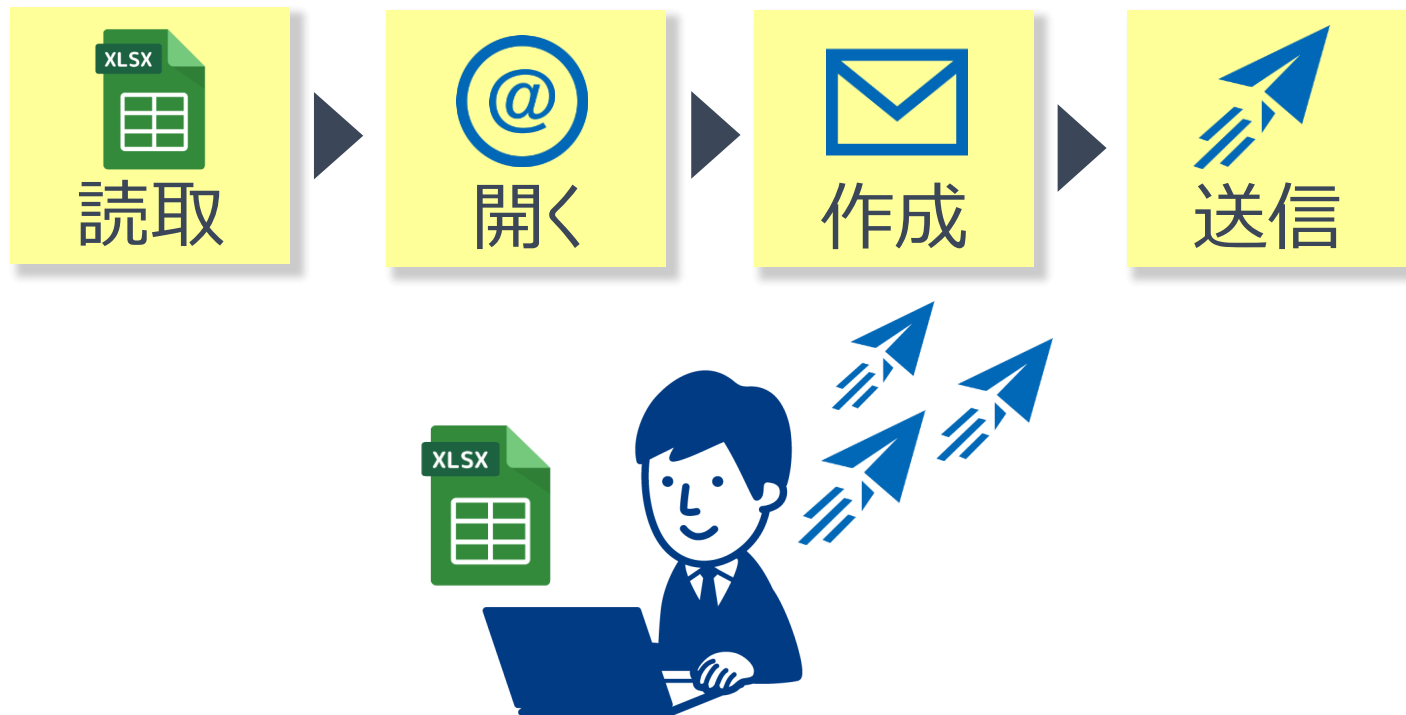


シナリオの実行と保存は、“Webページのクリック & 入力を自動化する”の3-11、3-12と同様に行います。

9-4 シナリオの実行

9-5 シナリオの保存

基礎編



メール送信を自動化する

導入

台帳作成も無事にこなし、社内でWinActorユーザーとして認知されはじめました。
そんな中、今度は今までとは違う頼まれごとが来ました。

?



「経理部から、支払処理台帳に記載がある担当者にメールを送りたいと相談されてしまった。
メールを送信するにはライブラリにある＜メール送信＞を使えばいいのかな？」



「宇院さんはすっかりWinActorに慣れてきましたね。もちろんメール送信の自動化もできます。
今回も既存の部品を組み合わせで自動化をしていきましょう。」



「メールは誤送信ひとつで事故になってしまうため自動化できるとセキュリティ的にも安心です。」



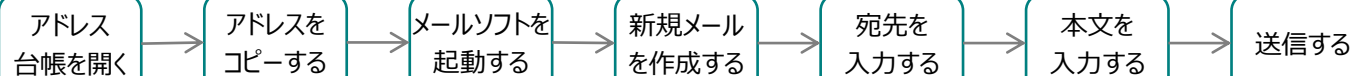
「今回も手作業とWinActorを使った自動化とで業務がどう変わるか確認しましょう。」



「WinActorを使った自動化では以下のように業務が変わります。」

手
作
業

人

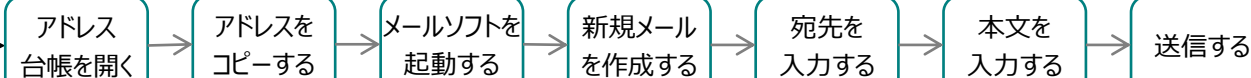


自
動
化

人

WinActor
起動

Win
Actor



「アドレス帳を開いてアドレスをコピーするところまでは今までの経験で出来そうです！」



「今回は、＜Outlook メール送信＞というライブラリを利用します。」



「メールソフトも扱えるんですね！」

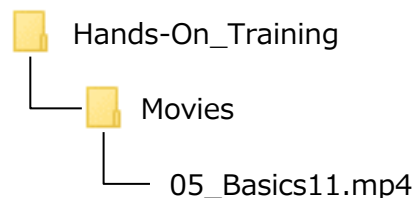
「はい、Outlook メール送信はOutlookでメールを送信できるライブラリになります。宛先や件名、本文も自由に設定できますよ！」



「メール送信は、シナリオの実行結果を利用者に通知したり、シナリオが止まってしまった際にエラーを通知したりするためなど幅広い使用方法があります。しっかりと利用方法を学んでおきましょう！」



「それでは実際のWinActorの動きを見てみましょう。以下フォルダのファイル名をダブルクリックして再生してください。」



「手動でメールを送信する際のように新規メールウィンドウは表示されませんが、Outlookの中でメール送信の処理がされています。」



「それでは実際にシナリオを一緒に作っていきましょう！」

11

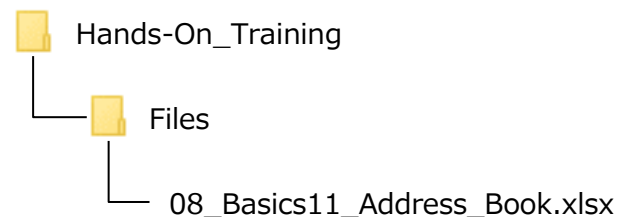
メール送信を自動化する

準備

アドレス帳の確認

	A	B	C	D	E	F	G
1							
2		アドレス帳					
3							
4		通番	部署	氏名	メールアドレス	備考	
5		1	経理部	鈴木 太郎	suzukit@ntttest.com		
6		2	経理部	手塚 雅生	teduka@ntttest.com		
7		3	経理部	河野 保二	kawano@ntttest.com		
8		4					
9		5					
10		6					
11		7					
12		8					
13		9					
14		10					
15		11					

確認するファイル



今回はこのアドレス帳からメールアドレスを取得します。メールアドレスは送信できるアドレスに修正してください。

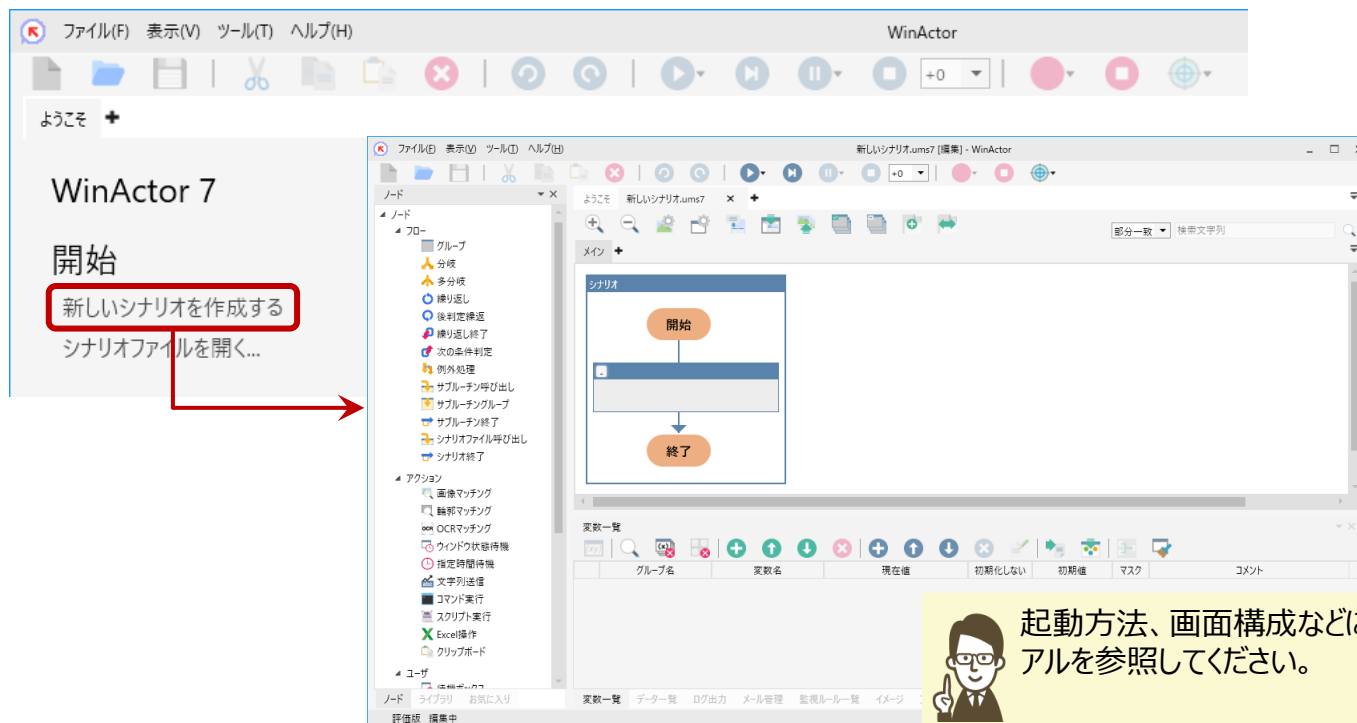
11

メール送信を自動化する

準備

WinActorの起動

- 1 スタートメニューなどからWinActorを起動します。
- 2 「新しいシナリオを作成する」をクリックすると、WinActorのメイン画面が表示されます。



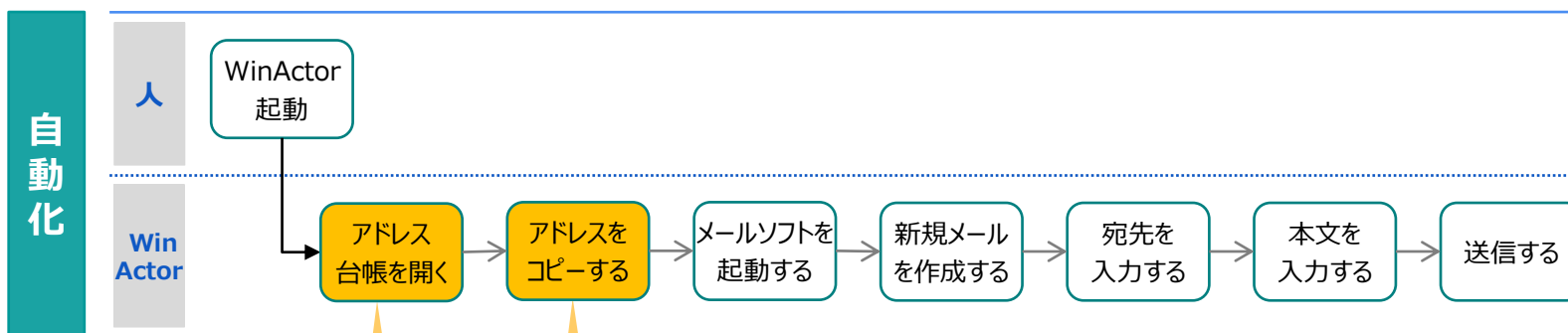
起動方法、画面構成などについては操作マニュアルを参照してください。

11

メール送信を自動化する



「これから以下の業務についてシナリオを作成していきます。」

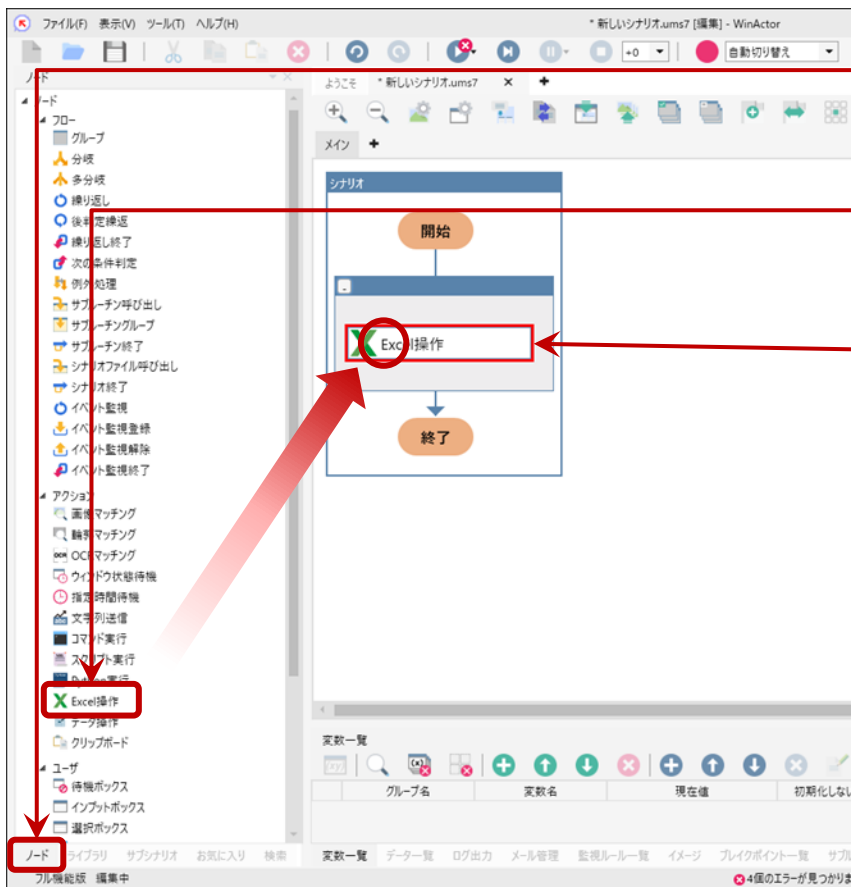


11-1 Excelからの値読み込み（ノード利用）

11

メール送信を自動化する

11-1 Excelからの値読み込み（ノード利用）



1 <ノード> をクリックします。

2 <Excel操作> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したノードをダブルクリックします。(プロパティを表示)

11

メール送信を自動化する

11-1 Excelからの値読み込み（ノード利用）

プロパティ

Excel操作

名前 Excel操作

コメント

操作 値の取得

取得元

ファイル名 値⇒ C:\temp\08_Basics11_Address_

シート名 値⇒ Sheet1

セル位置 値⇒ E5

格納先

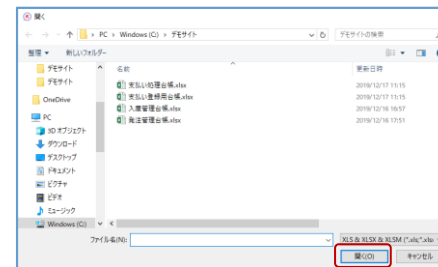
変数 メールアドレス

更新

元に戻す

4 <操作> のプルダウンリストから<値の取得> を選択します。

5 <ファイル名> 参照ボタンをクリックすると、エクスプローラーが開きます。参照先となる「08_Basics11_Address_Book.xlsx」を指定し、開きます。



6 <シート名> に、「値⇒ Sheet1」（参照ボタンも利用できます）
<セル位置> に、「値⇒ E5」を入力します。

7 <変数> をクリックし、「メールアドレス」と入力します。

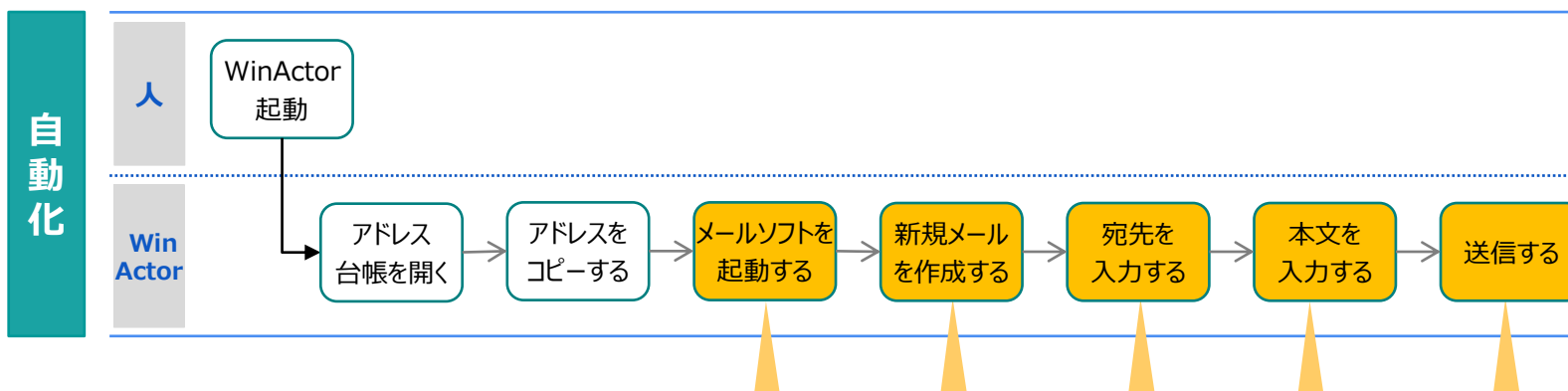
8 <更新> ボタンをクリックします。変数登録確認のダイアログは<はい> をクリックします。

11

メール送信を自動化する



「これから以下の業務についてシナリオを作成していきます。」

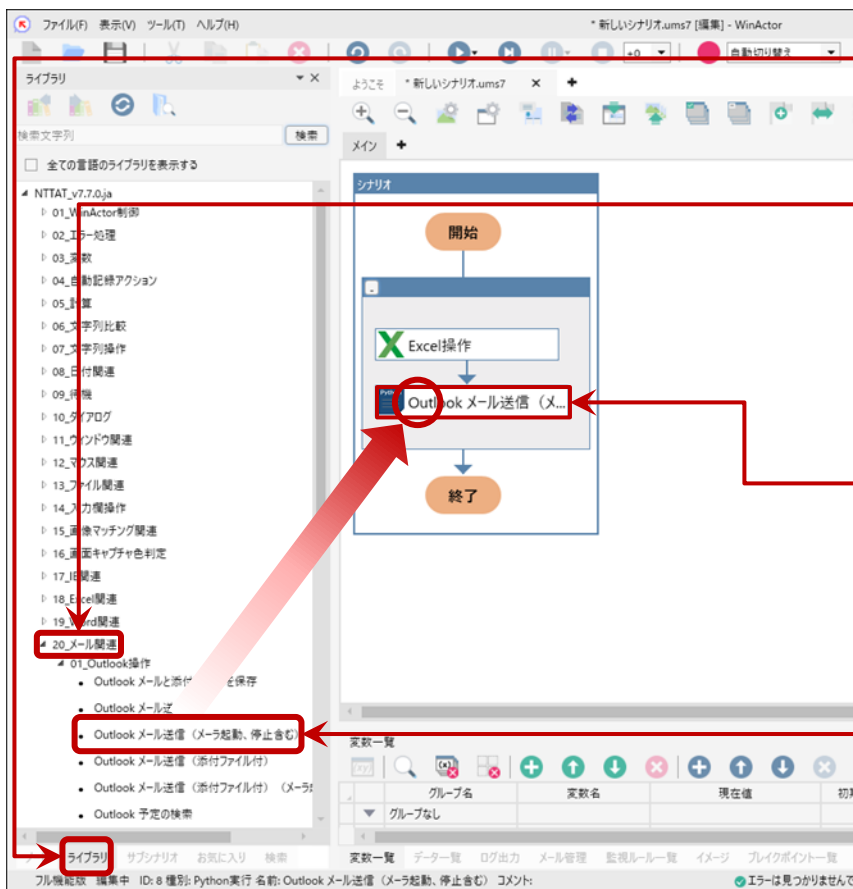


11-2 メールの送信（ライブラリ利用）

11

メール送信を自動化する

11-2 メール送信 (ライブラリ利用)



1 <ライブラリ> をクリックします。

2 <20_メール関連> をダブルクリックして、リストを展開します。

3 <Outlook メール送信 (メーラ起動、停止含む)> をドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

Outlookでメールを送信するにはいくつかの方法があります。

今回はメール送信だけでなく、Outlookの起動と停止も行ってくれる<Outlook メール送信 (メーラ起動、停止含む)>を使用します。作成するシナリオによっては、ファイルを添付できる<Outlook メール送信 (添付ファイル付) (メーラ起動、停止含む)>などをご利用ください。



11

メール送信を自動化する

11-2 メール送信 (ライブラリ利用)

プロパティ

Python実行

名前 Outlook メール送信 (メーラ起動、停止含む)

コメント

設定 スクリプト 注釈 バージョン情報

件名 :
送信するメールの件名を指定してください。

本文 :
送信するメールの本文を指定してください。

宛先 (To) 値⇒

宛先 (Cc) 値⇒

件名 値⇒

本文 値⇒

更新 元に戻す

5

<宛先 (To)> に、「メールアドレス」
<宛先 (Cc)> は、入力不要
<件名> に、「値⇒ テストメール」
<本文> に、「値⇒ テストメール本文」
と入力します。

6

<更新> ボタンをクリックします。

<Outlook メール送信 (メーラ起動、停止含む)> ライブラリは、メール送信以外にOutlookの起動と停止も含まれている便利なライブラリです。しかし、Outlookの設定にてメールを作成後すぐに送信する設定がされていないと、Outlook停止時に確認ダイアログが表示され正常に停止が出来ません。利用する際はご注意ください。

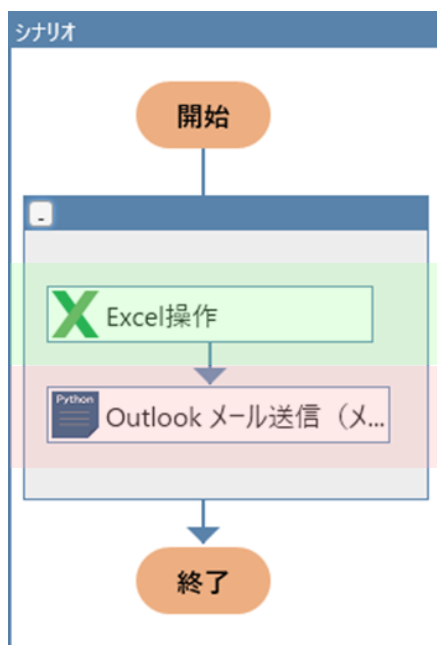


11

メール送信を自動化する

11-3 シナリオ全体の確認

1 作成したシナリオが以下の通りになっていることを確認します。



アドレス帳から
アドレスを取得(11-1)

取得したアドレスを参照し
メールを送信(11-2)

ライブラリを利用することで簡単にシナリオを作成することができます。WinActorではよく使われる業務をすぐに自動化できるようにライブラリがたくさん用意されているので、無理にノードだけでシナリオを作らず、ライブラリの利用も考えてみましょう。



シナリオの実行と保存は、“Webページのクリック & 入力を自動化する”の3-11、3-12と同様に行います。



11-4 シナリオの実行

11-5 シナリオの保存



「宇院さん、基礎編が一通り終わりましたがどうでしょうか？」



「基礎編が終わった自分が言うのも何ですが、思った以上にカンタンに自動化ができました！
色々なノードやライブラリが豊富にあって、ほとんどドラッグ＆ドロップでシナリオは作れますし、
作成したシナリオをすぐに動かして、動かないところはすぐに直してまた動かすサイクルができる
ことがあきらめずにできたコツですかね！」



「それはよかったです！次は、繰り返し処理や分岐処理など実際の業務でWinActorを使用する際にマスターしておくべき機能を学ぶ実践編です。ここで満足せず次のステップに進みましょう！」

～ 実践編 ～



1

シナリオ作成 ～実践編プロローグ～

導入

宇院さんはいくつかWinActorでシナリオを作り自動化を進めてきましたが、もう少し複雑な処理をしたいようです。

今回も宇院さんと一緒に少し複雑な自動化を作ってみましょう!!



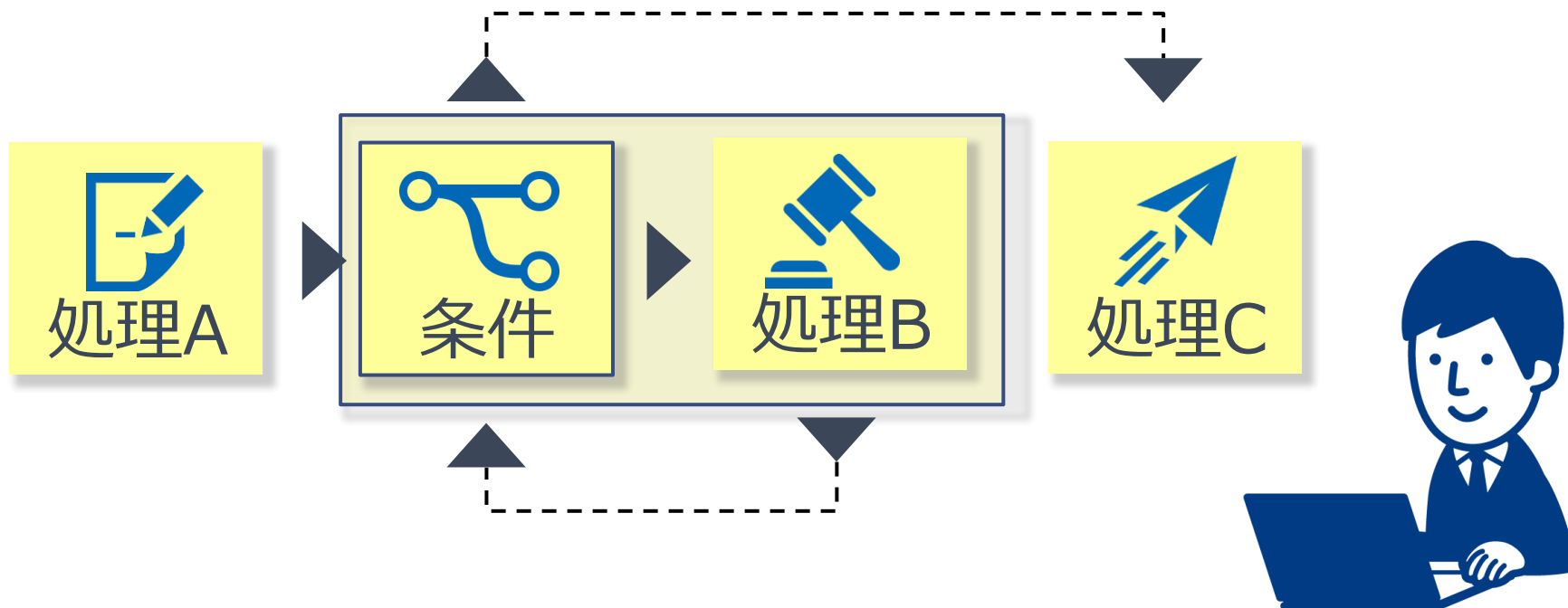
宇院(ういん)さん

調達部で働く2年目社員。
WinActorの使いどころが分かり、他の業務にも使用していきたいと思っている。



増田(ますた)先輩

既にWinActorを使用している他部署の先輩社員。
今回宇院さんの相談を受け、サポート役をかって出た。



同じ処理を繰り返し行う

導入

宇院さんのもとに、財務部から数件の支払い情報をまとめて社内システムに登録したいという依頼が来ました。宇院さんは財務部が納得するシナリオを作ることができるのでしょうか。



「ダメだ…今までの経験では同じ作業を繰り返す動きができない…＜繰り返し＞のノードを使えばいいと思うんだけど、そもそも他のノードと形が違うしどう使えばいいんだ…？」



「今まで宇院さんが作ってきたシナリオは、開始から終了まで動いて終わりでしたからね。WinActorが得意な、大量のデータを繰り返し処理する方法を学んでいきましょう。」



「しかし、そもそもどのように作ればいいのか検討もつかないのです…。件数が一定ならば同じ処理を件数分繋げればいいのですが、毎回件数は違うのです。」

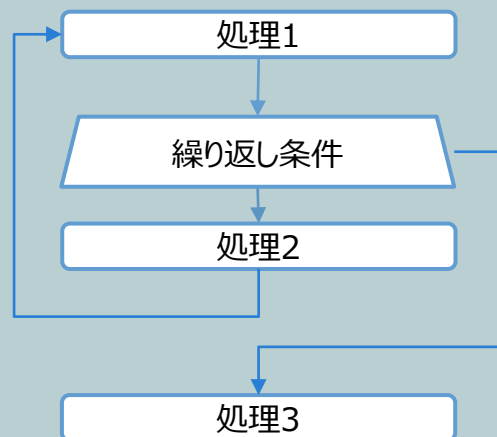


「大丈夫です！そのためにまずは繰り返し処理の概念から学びましょう。」



「そもそも繰り返し処理って何ですか？」

「**同じ処理を、事前に設定した処理終了条件が満たされるまで繰り返すこと**を言います。日ごろ宇院さんが行っている発注処理も、発注台帳に記載されている件数の処理が終われば作業は終了になりますよね？これと同じことをWinActorで処理することができます。」



■ 繰り返し条件
入力された値が
「10」になるまで
処理2を繰り返
しを実施する

「左図は繰り返し処理のイメージです。条件を満たさない場合は処理2を行い再度条件判定を行います。満たす場合は次の処理3に進みます。」

「この条件をどう設定するかが、繰り返し処理のポイントになります。」



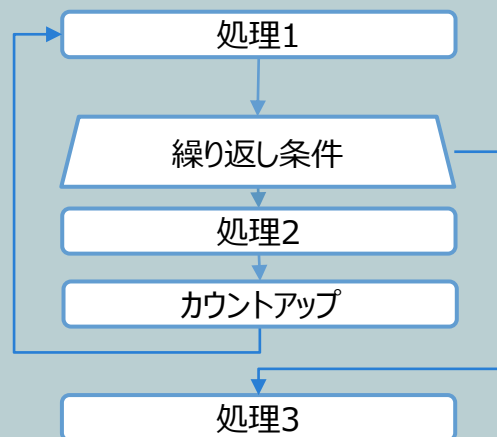
「何となく分かってきました！でも同じ処理を繰り返しても永遠に同じ内容が処理されるだけなんじゃないですか？私は台帳に書かれている値を1行ずつ下に移動しながら登録をしたいんですが…」

「その通りです、そこで繰り返し処理のもう一つのポイントである**カウントアップ**という概念を説明します。」

「**カウントアップ**とは繰り返し処理のたびに設定した変数の数値が一定の数ずつ加算されていくことです。これによって繰り返し終了の回数を設定したり、Excelの行数を1行ずつずらして取得できます！」



Point!



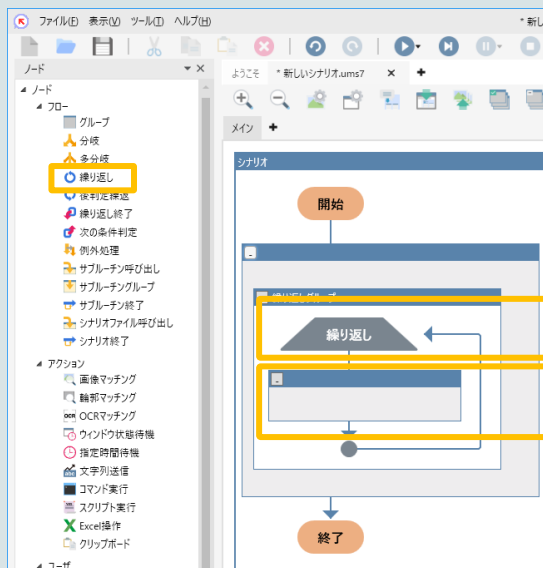
■ 繰り返し条件
入力された値が
「10」になるまで
処理2を繰り返
しを実施する

「左図はカウントアップを追加した繰り返し処理のイメージです。カウントアップをすることによって、処理2を決まった値だけでなく繰り返し回数によって違う値を使用することができるようになります。」
「条件の設定方法とカウントアップを学ぶことで繰り返し処理をマスターすることができます！」



「台帳で1行ずつ下に移動するために1ずつ加算されるカウントアップの考え方を使うのですね！実際にWinActorで設定するためにはどうすればいいのですか？」

「画面左上のノードタブで表示される、＜繰り返しノード＞を使用します。」
「繰り返しノードは、繰り返しの終了条件を決める台形の繰り返し条件(※1)と、何の処理を繰り返すか設定する四角のエリア(※2)の二つで出来ています。」



「条件と処理内容を設定することで、繰り返し処理ができます。条件は台形をダブルクリックして表示される詳細設定から設定を行い、処理内容は他のシナリオと同じようにノードやライブラリをドラッグ&ドロップすることで処理を設定することができます。」

Point!



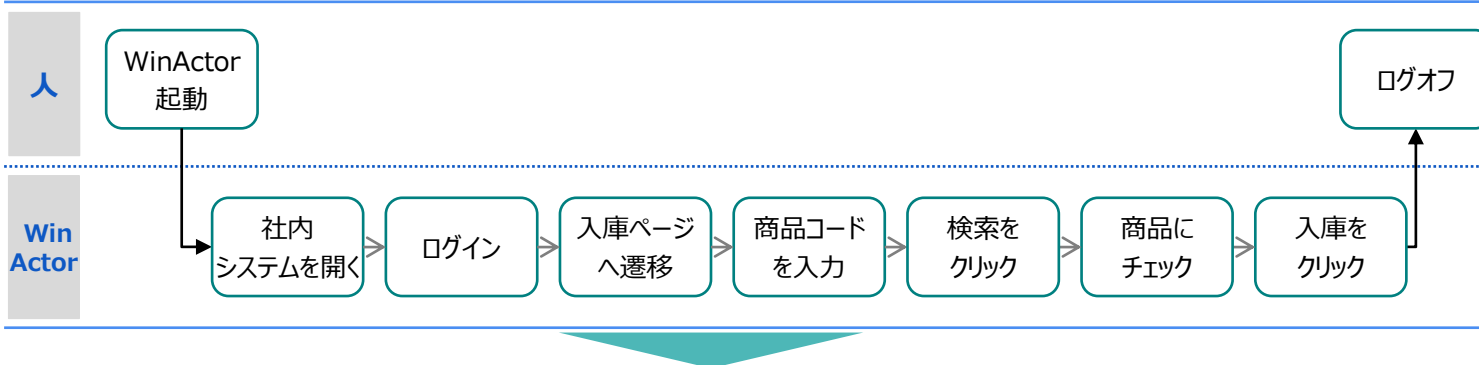
2

シナリオ作成 ～実践編.1～

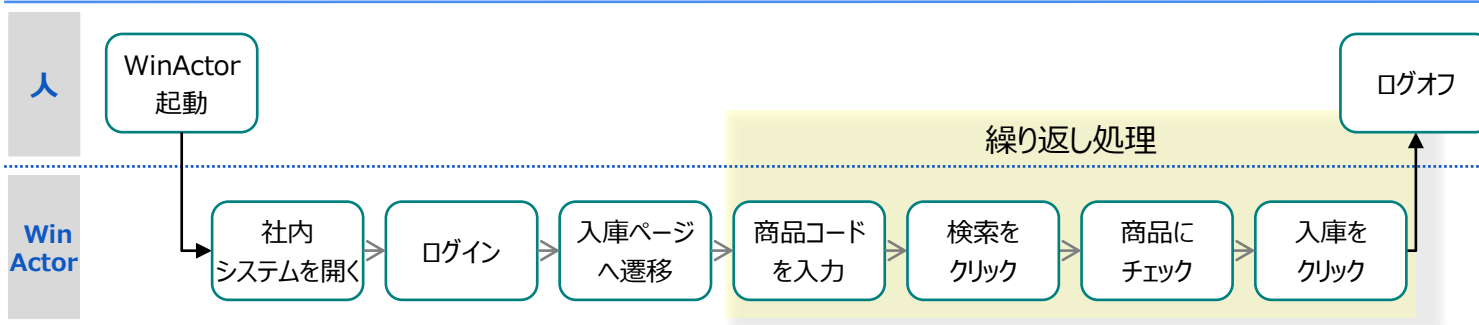


「それでは、基礎編で作成した発注登録を行うシステムに繰り返し処理を追加してみましょう。」

処理の流れ



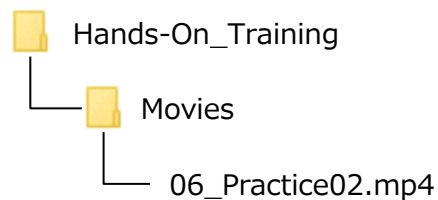
処理の流れ



「事前にどこを繰り返し処理するか決めておくと、シナリオ修正がスムーズに進みますね！」



「それでは実際のWinActorの動きを見てみましょう。以下フォルダのファイル名をダブルクリックして再生してください。」



「それでは実際にシナリオを一緒に作っていきましょう！」

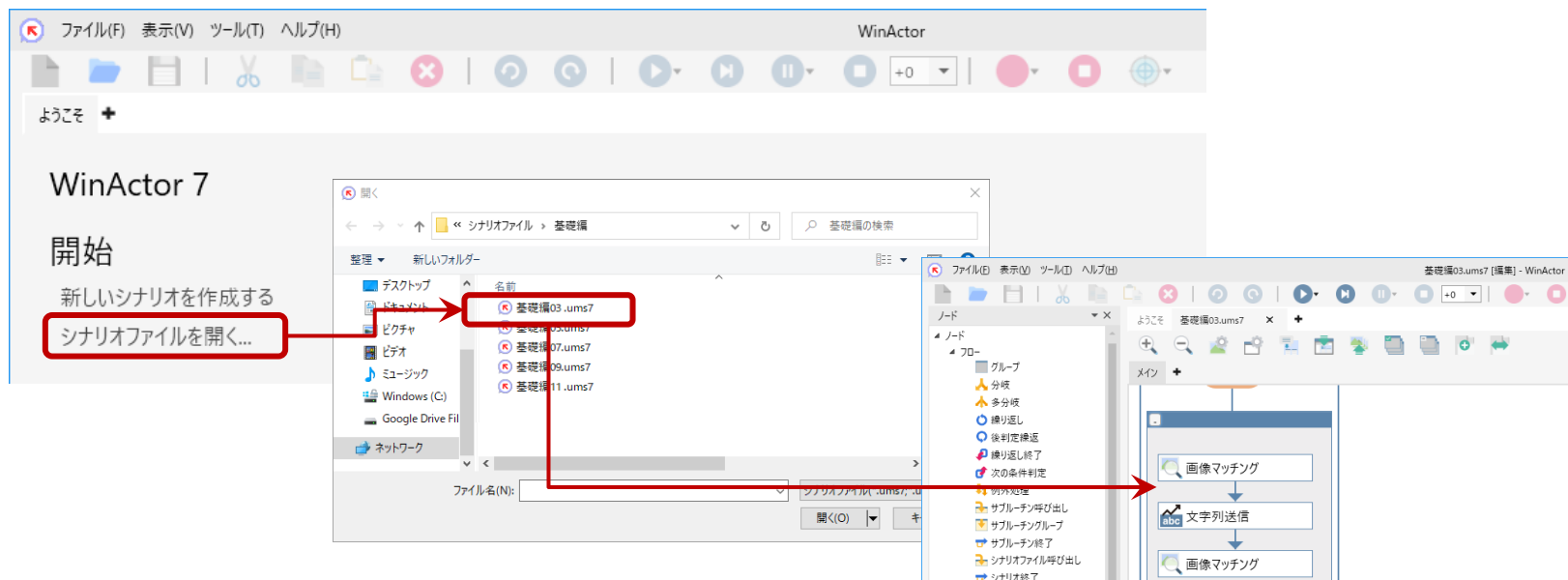
3

何度も同じ処理を繰り返し行う

準備 WinActorの起動、既存シナリオを開く

1 スタートメニューなどからWinActorを起動します。

2 「シナリオファイルを開く…」をクリックし、基礎編「Webページのクリック&入力を自動化する」で保存したシナリオを選択すると、WinActorのメイン画面に作成したシナリオが表示されます。



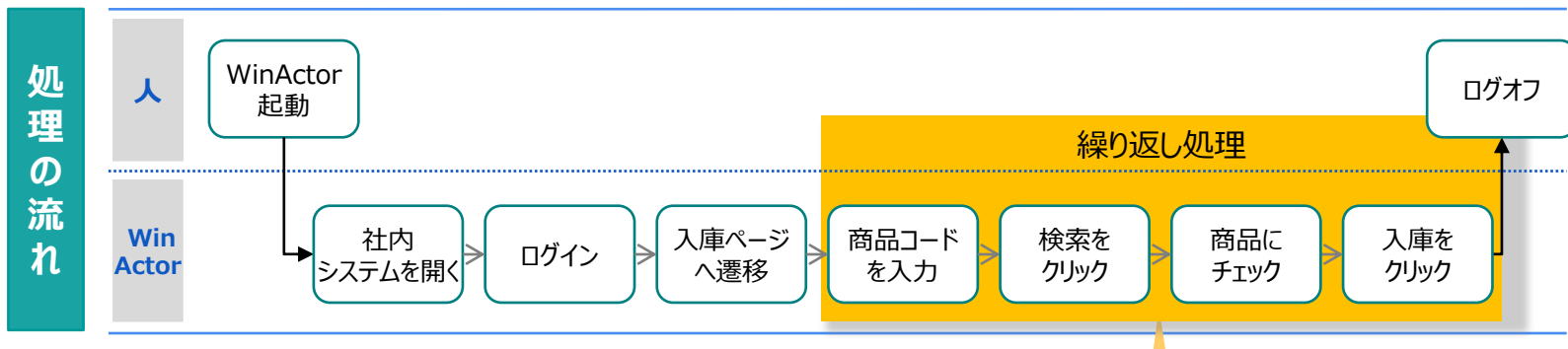
起動方法、画面構成などについては操作マニュアルを参照してください。

3

何度も同じ処理を繰り返す行う



「これから以下の業務についてシナリオを作成していきます。」



準備 ログイン画面の表示

準備 追加処理の確認

3-1 繰り返し回数の設定

3-2 台帳の入力開始行設定

3-3 繰り返し処理の設定

3-4 値取得設定

3-5 値入力設定

3

何度も同じ処理を繰り返し行う

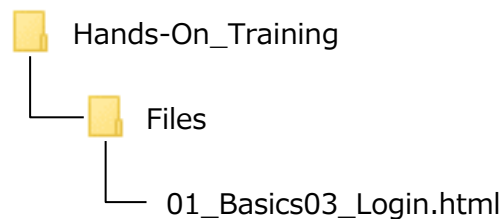
準備

ログイン画面の表示

- 1 「01_Basics03_Login.html」を右クリックして、「プログラムから開く」から「Microsoft Edge」を選択します。



使用するファイル



デモシステムを起動するブラウザーには
Edge または Chrome を使用してください。



Chromeを使う人は「プログラムから開く」から
「Google Chrome」を選択してください。

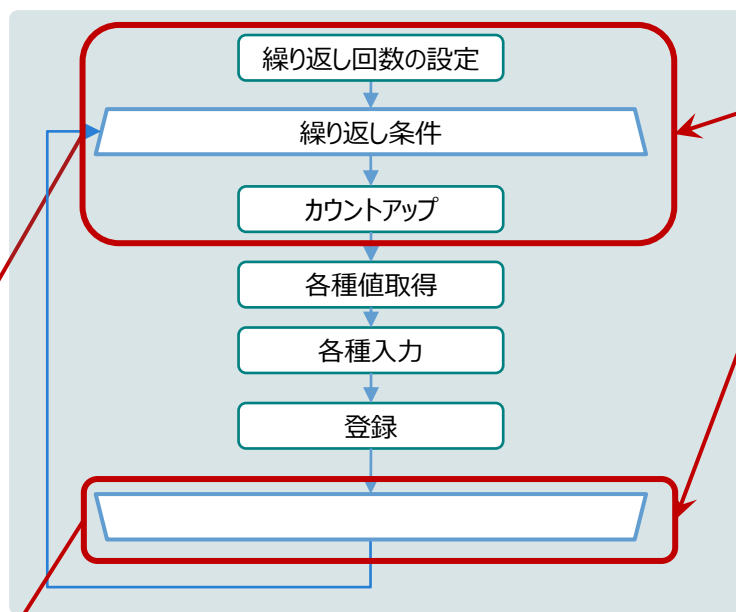
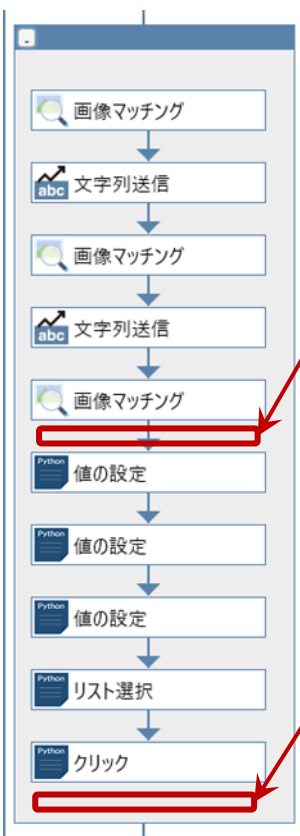
3

何度も同じ処理を繰り返し行う

準備

追加処理の確認

1 今回は以下の処理を追加します。



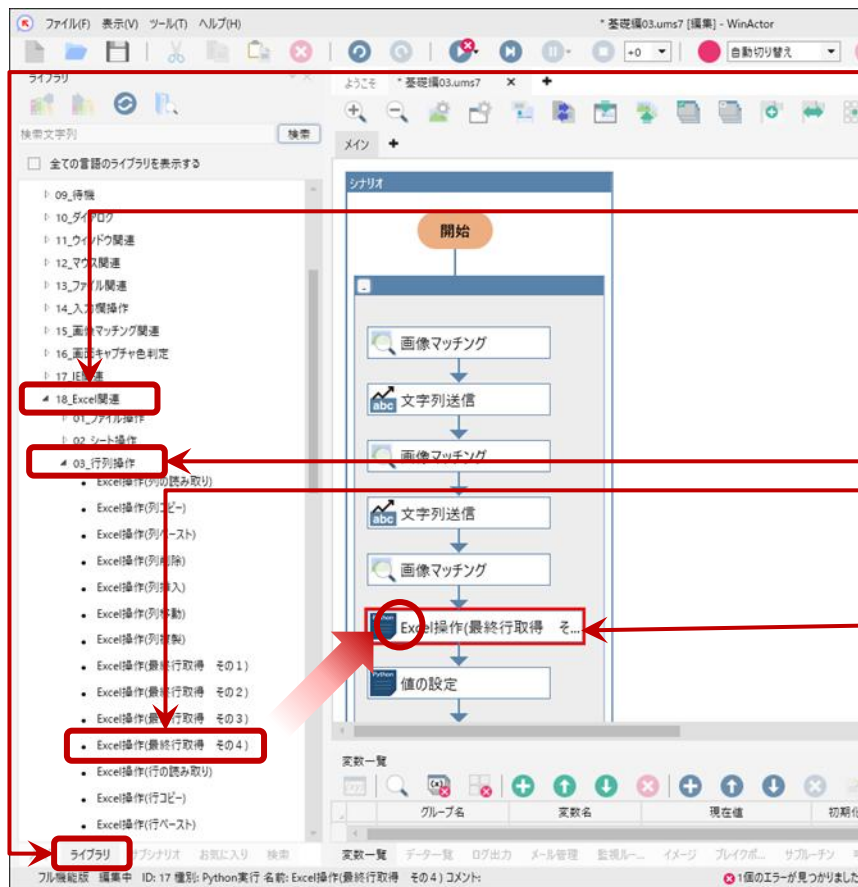
2

- ・繰り返し回数の設定
- ・繰り返し処理の設定

3

何度も同じ処理を繰り返し行う

3-1 繰り返し回数の設定



1 <ライブラリ> をクリックします。

2 <18_Excel関連> をダブルクリックして、リストを展開します。

3 <03_行列操作> をダブルクリックして、リストを展開します。

4 <Excel操作(最終行取得 その4)> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

5 配置したライブラリをダブルクリックします。(プロパティを表示)

最終行の行数を取得した後、不要な行を削除し繰り返し回数を設定します。
まずは台帳のすべての情報を取得したいので、最後の行まで取得してくれる<Excel操作(最終行取得 その4)>というライブラリを活用します。



3

何度も同じ処理を繰り返し行う

3-1 繰り返し回数の設定

プロパティ Python実行

名前 Excel操作(最終行取得 その4)

コメント

設定 スクリプト 注釈 バージョン情報

[ファイル名]: 操作対象のファイルパスを指定してください。
相対パス、絶対パス、またはURLが利用可能です。

[シート名]: 操作対象のシート名を指定してください。
空の場合はアクティブなシートが選択されます。

[検索列]: 最終行を検索する列を指定してください。
例) A

[最終行]: 取得した最終行を格納する変数を指定してください。

ファイル名 値⇒ C:\temp\04_Basics05_Hacchu_Ka

シート名 値⇒ Sheet1

検索列 値⇒ C

最終行 最終行

更新 元に戻す

6

<ファイル名> に、
04_Basics05_Hacchu_Kanri_Daicho.xlsx をドラッグ&ドロップし
<シート名> に、「値⇒ Sheet1」
<検索列> に、「値⇒ C」
<最終行> に、「最終行」
を入力します。

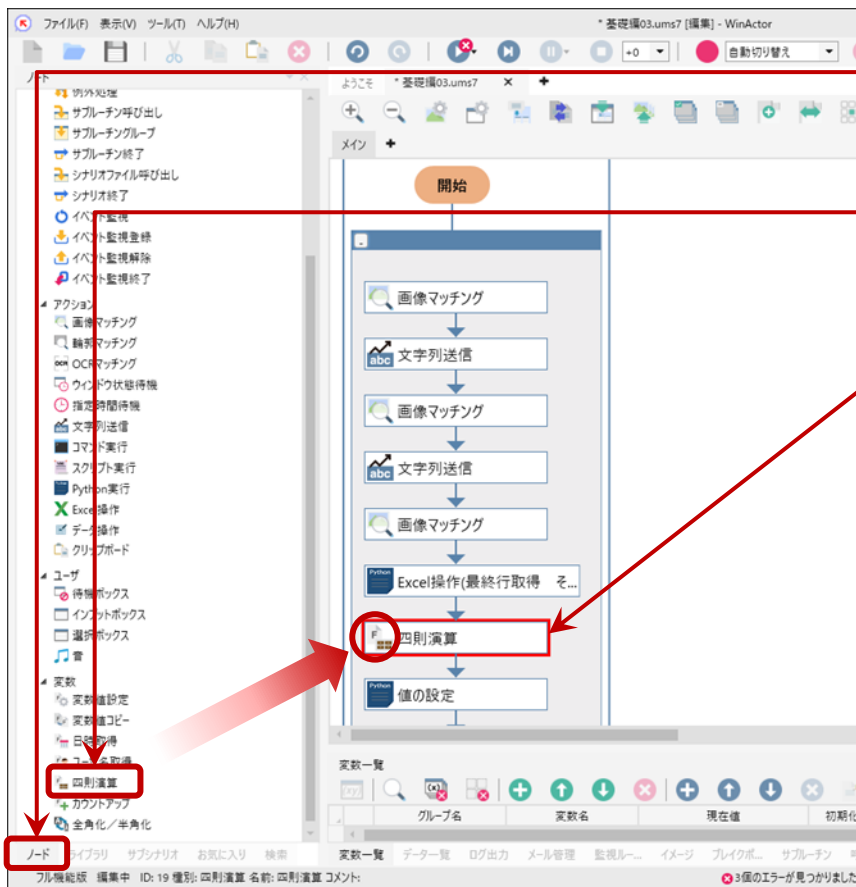
7

<更新> ボタンをクリックします。変数登録確認のダイアログは <はい> をクリックします。

3

何度も同じ処理を繰り返し行う

3-1 繰り返し回数の設定



8 <ノード> をクリックします。

9 <四則演算> ノードをドラッグし、シナリオ編集エリア上でドロップします。

10 配置したノードをダブルクリックします。(プロパティを表示)

3

何度も同じ処理を繰り返し行う

3-1 繰り返し回数の設定

11 <計算結果>に「繰り返し回数」を入力します。

12 左の<変数名もしくは値を選択>のプルダウンから、「最終行」を選択、中央の<演算子>のプルダウンから、「-（マイナス）」を選択、右の<変数名もしくは値を選択>のプルダウンから、「値⇒ 4」を入力します。

13 <更新> ボタンをクリックします。変数登録確認のダイアログは<はい>をクリックします。

通番	注文日	承認日	依頼部署
1	2019/12/25	2019/12/25	経理部
2	2019/12/28	2019/12/25	経理部
3	2019/12/29	2019/12/25	経理部
4			
5			

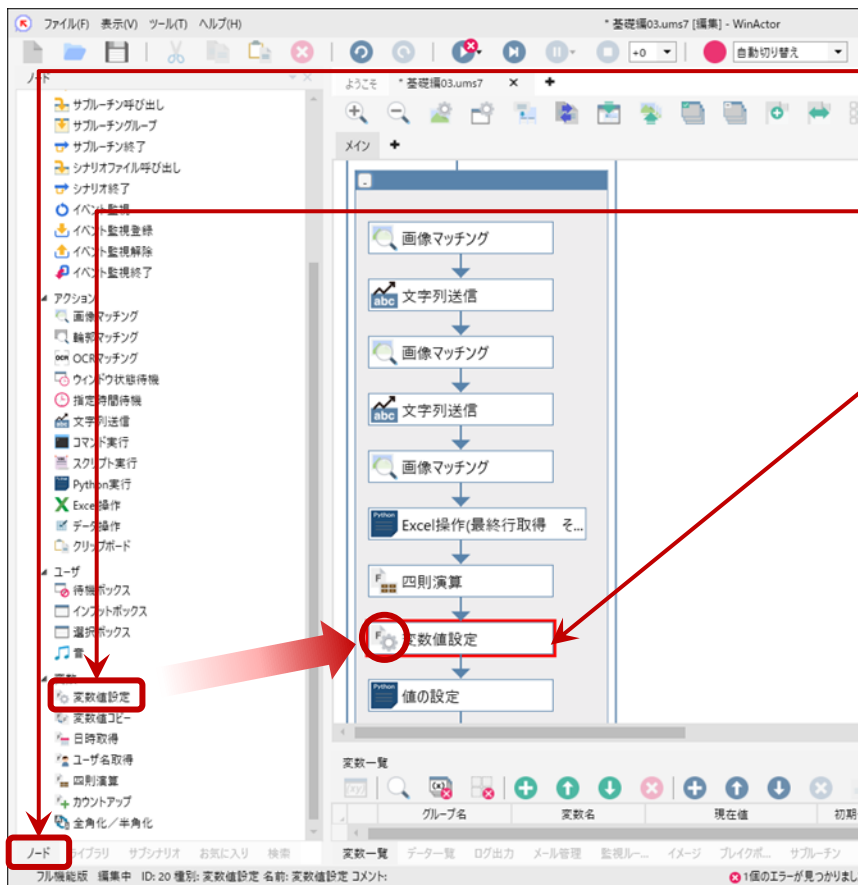


4を引くのは、台帳上には台帳名など実際に入力する値以外の行があるためです。
今回は5行目から実際に入力したい値になっているため、正しい繰り返し回数に調整しています。

3

何度も同じ処理を繰り返し行う

3-2 台帳の入力開始行設定



1 <ノード> をクリックします。

2 <変数値設定> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したノードをダブルクリックします。(プロパティを表示)

3

何度も同じ処理を繰り返し行う

3-2

台帳の入力開始行設定

プロパティ

変数値設定

名前 変数値設定

コメント

変数名 カウンター

設定値 4

更新

元に戻す

4 <変数名> に、「カウンター」を入力します。

5 <設定値> に、「4」を入力します。

6 <更新> ボタンをクリックします。変数登録確認のダイアログは<はい> をクリックします。



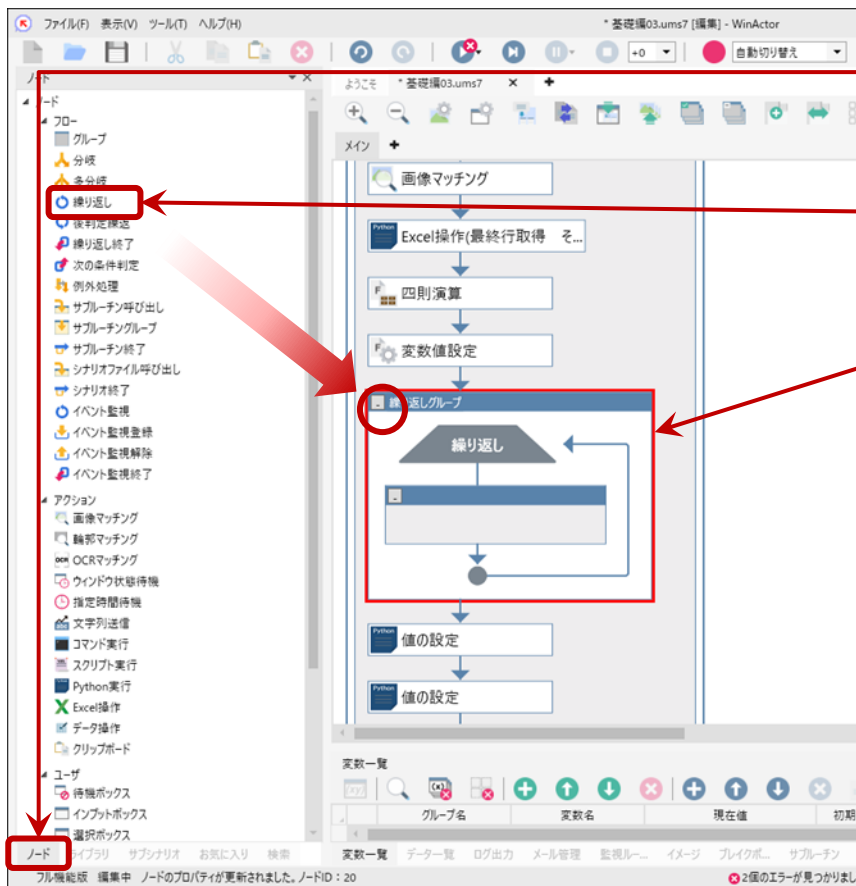
カウンター変数にあらかじめ4を設定しておくのは、台帳の5行目から入力を行いたいからです。繰り返し処理の中の初めにカウンターに+ 1を行うため4と設定しています。

3

何度も同じ処理を繰り返し行う

3-3

繰り返し処理の設定



1 <ノード> をクリックします。

2 <繰り返し> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したノードをダブルクリックします。(プロパティを表示)

3

何度も同じ処理を繰り返し行う

3-3

繰り返し処理の設定

繰り返し

名前 繰り返しグループ

コメント

☐ 条件式

☒ 回数 繰り返し回数

☐ 範囲 変数名もしくは値を選択 から 変数名もしくは値を選択

☐ データ数 データファイル名 変数名もしくは値を選択

☐ データ数 (DB連携) データソース名 変数名もしくは値を選択

ユーザ名 変数名もしくは値を選択

パスワード 変数名もしくは値を選択

テーブル名 変数名もしくは値を選択

カウンタ 変数名を選択 (省略可)

4

＜回数＞のラジオボタンをクリックし、「繰り返し回数」を選択します。

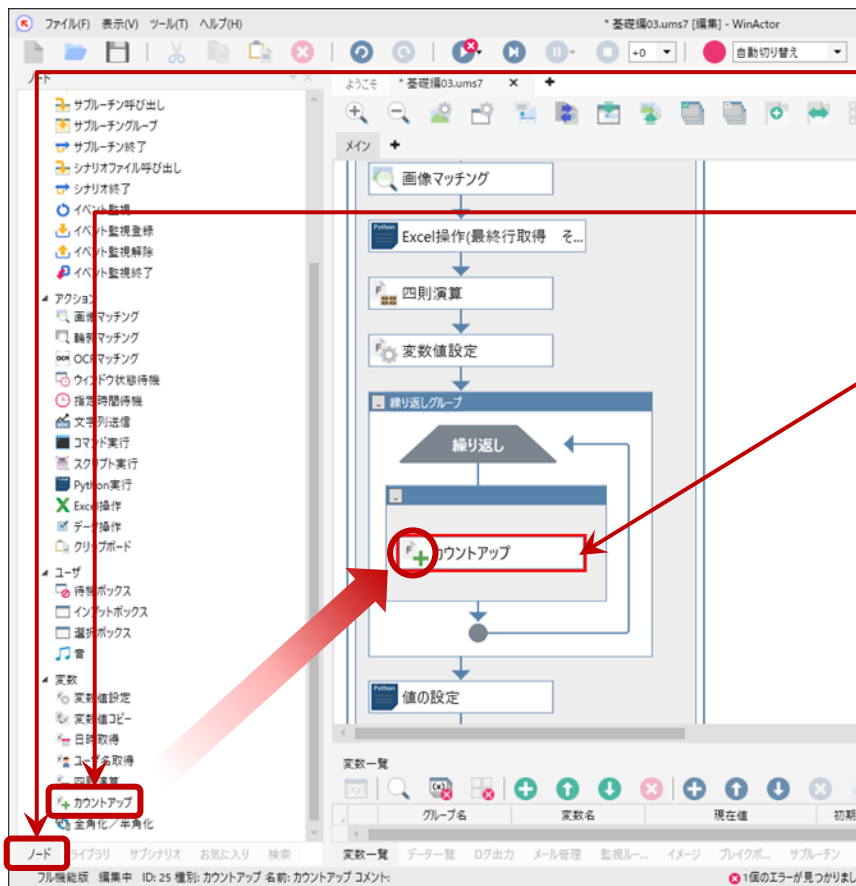
5

＜更新＞ボタンをクリックします。

3

何度も同じ処理を繰り返し行う

3-3 繰り返し処理の設定



6 <ノード> をクリックします。

7 <カウントアップ> ノードをドラッグし、シナリオ編集エリア上でドロップします。

8 配置したノードをダブルクリックします。(プロパティを表示)

3

何度も同じ処理を繰り返し行う

3-3

繰り返し処理の設定

カウントアップ

名前 カウントアップ

コメント

計算結果 カウンター

加算値 1

更新 元に戻す

9

＜計算結果＞のプルダウンから、「カウンター」を選択します。

10

＜加算値＞に、「1」を選択します。

11

＜更新＞ボタンをクリックします。

3

何度も同じ処理を繰り返し行う

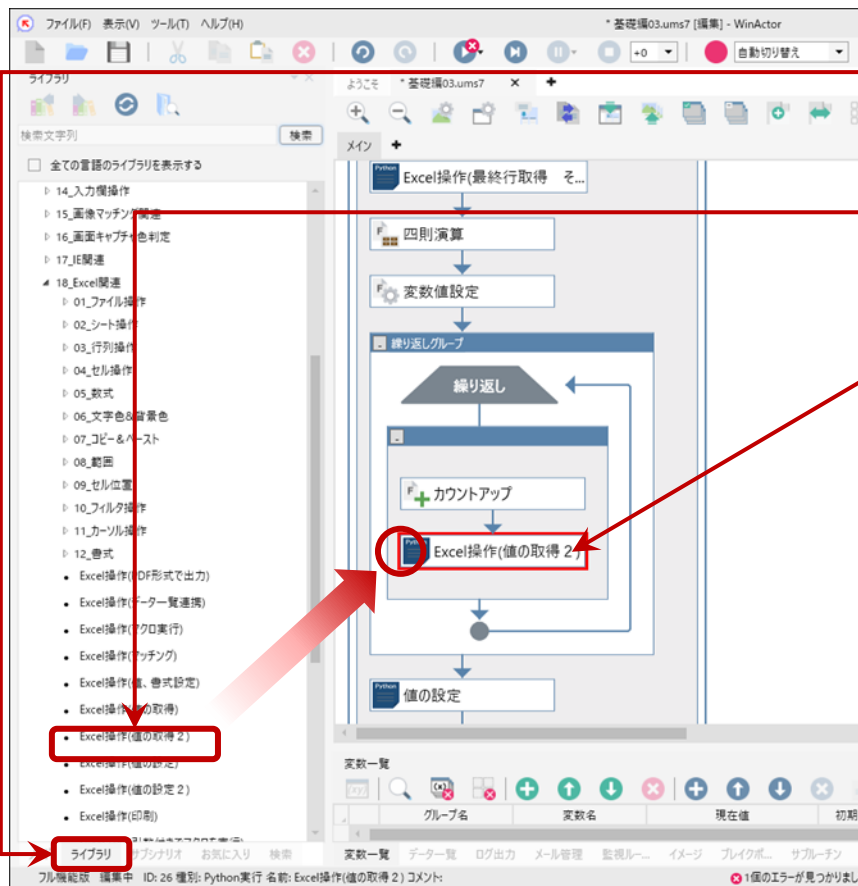
3-4

値取得設定

1 <ライブラリ> をクリックします。

2 <Excel操作(値の取得 2)> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したライブラリをダブルクリックします。(プロパティを表示)



3

何度も同じ処理を繰り返し行う

3-4

値取得設定

4

＜ファイル名＞に、
04_Basics05_Hacchu_Kanri_Daicho.xlsx をドラッグ
& ドロップし、
＜シート名＞に、「値⇒ Sheet1」
＜セル(行)＞に、「カウンター」
＜セル(列)＞に、「値⇒ C」
＜取得方法＞に、「value」
＜格納先変数＞に、「注文日」
を入力します。

5

＜更新＞ ボタンをクリックします。変数登録確認のダイアログ
は＜はい＞ をクリックします。

この設定で、＜セル(行)＞に、「カウンター」を
設定しているのがポイントです。



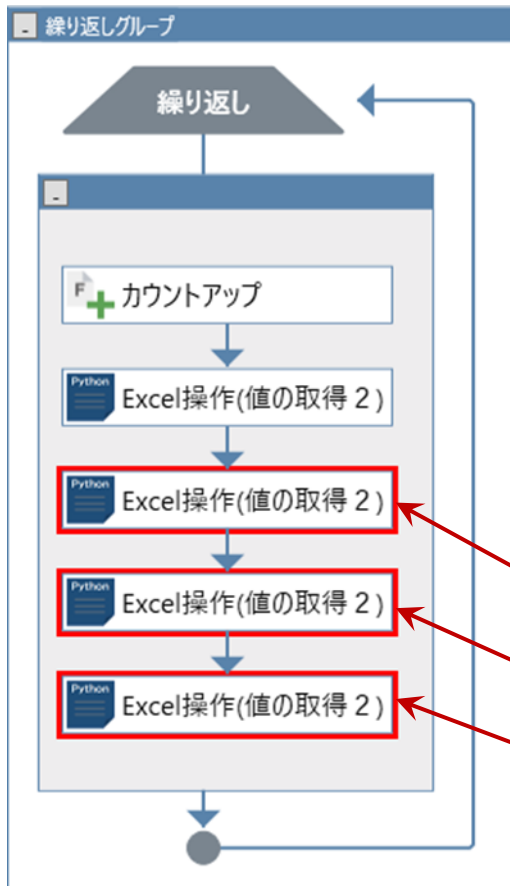
「カウンター」は繰り返しを行うごとに 1 ずつカウン
トアップされていくため、繰り返される度に行番号
が 1 ずつ増えていきます。

3

何度も同じ処理を繰り返し行う

3-4

値取得設定



前述した注文日を取得する設定のように、仕入先コード・担当者コード・商品コードについても設定します。
 <セル(列)>と<格納先変数>については下表を参考に設定ください。

6

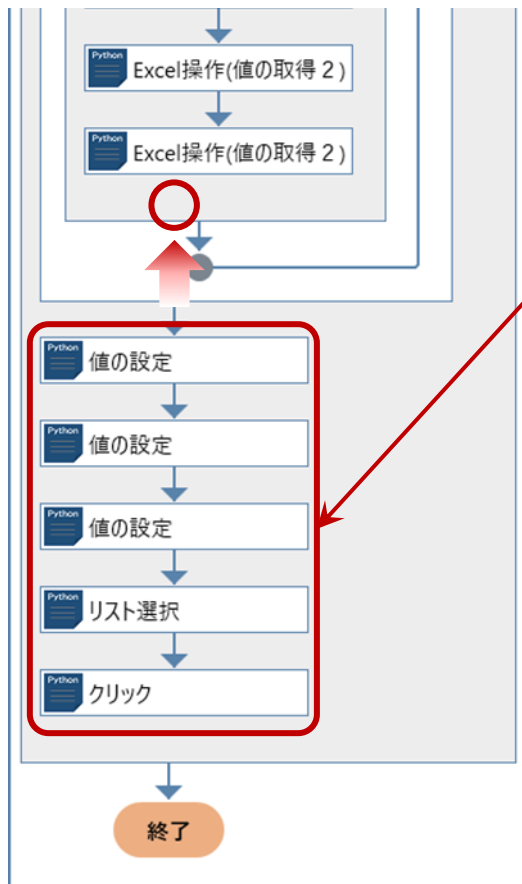
<ファイル名>、<シート名>、<セル(行)>、<取得方法>については、前述同様に、
 <ファイル名>に、
 04_Basics05_Hacchu_Kanri_Daicho.xlsx をドラッグ&ドロップし、
 <シート名>に、「値⇒ Sheet1」
 <セル(行)>に、「カウンター」
 <取得方法>に「value」を入力します。

セル(列)	格納先変数
G	仕入先コード
H	担当者コード
I	商品コード

3

何度も同じ処理を繰り返し行う

3-5 値入力設定



1

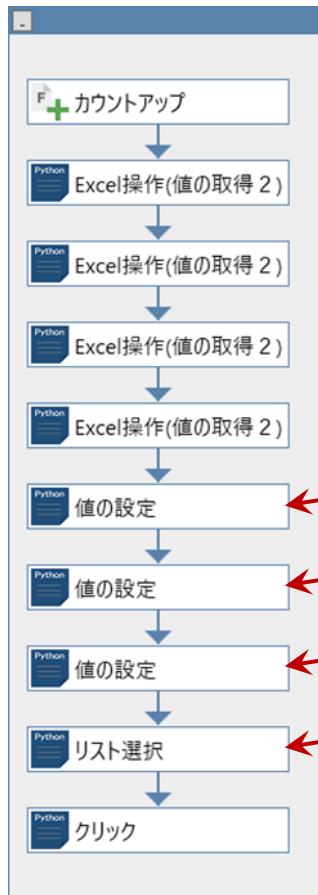
既存シナリオに設置されている、
<値の設定> 3つ と<リスト選択> 1つ と<クリック> 1つ を
繰り返し処理のグループに移動します。

3

何度も同じ処理を繰り返し行う

3-5

値入力設定



2

既存の注文日、仕入先コード・担当者コード・商品コードを入力するライブラリについて設定を変更します。
各ライブラリの〈設定する値〉または〈値〉について、プロパティを開き下表を参考に更新ください。

〈設定する値〉
または〈値〉

注文日
仕入先コード
担当者コード
商品コード

Tips

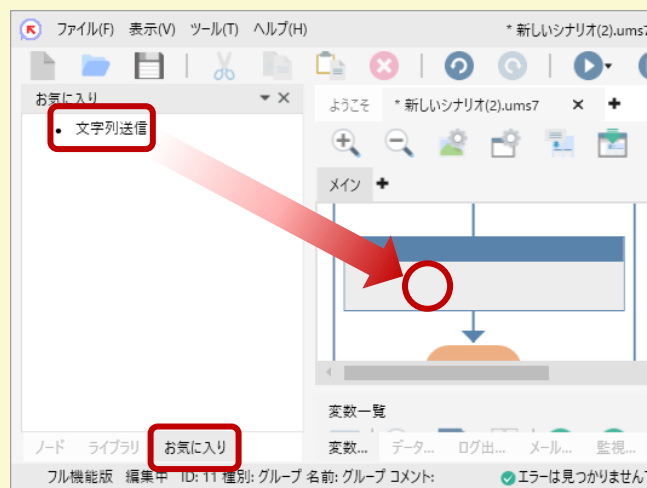
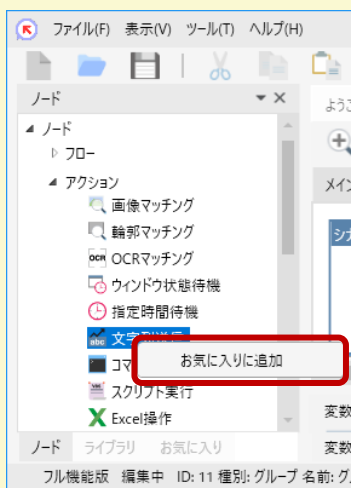
お気に入りノード、ライブラリ機能



よく利用するノードやライブラリは、お気に入りに設定しすぐに利用することができます。

よく利用するノードやライブラリを右クリックし、「お気に入りに追加」をクリックすることでお気に入りパレットに登録ができます。

登録されたノード、ライブラリは通常のノード、ライブラリと同じようにドラッグ＆ドロップで利用できるため、よく利用するノードやライブラリは登録しておき、ノード、ライブラリ選択の手間を減らしてシナリオ作成に弾みをつけましょう。



3

何度も同じ処理を繰り返し行う

3-6

シナリオ全体の確認

1 作成したシナリオが以下の通りになっていることを確認します。

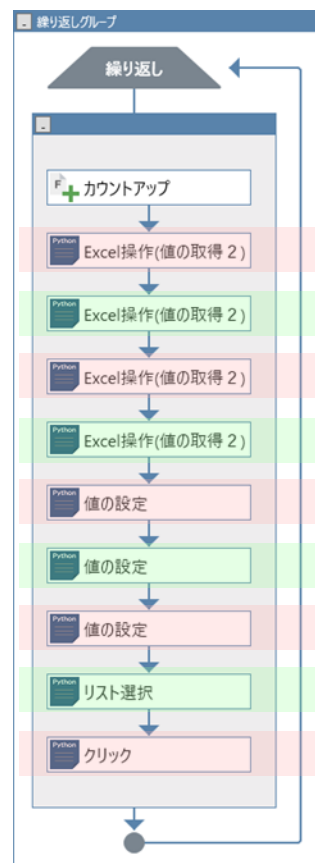


ユーザーID入力(既存)

パスワード入力(既存)

ログインボタンクリック(既存)

繰り返し回数設定(3-1)



注文日取得(3-3)

仕入先コード取得(3-3)

担当者コード取得(3-3)

商品コード取得(3-3)

注文日入力(3-4)

仕入先コード入力(3-4)

担当者コード入力(3-4)

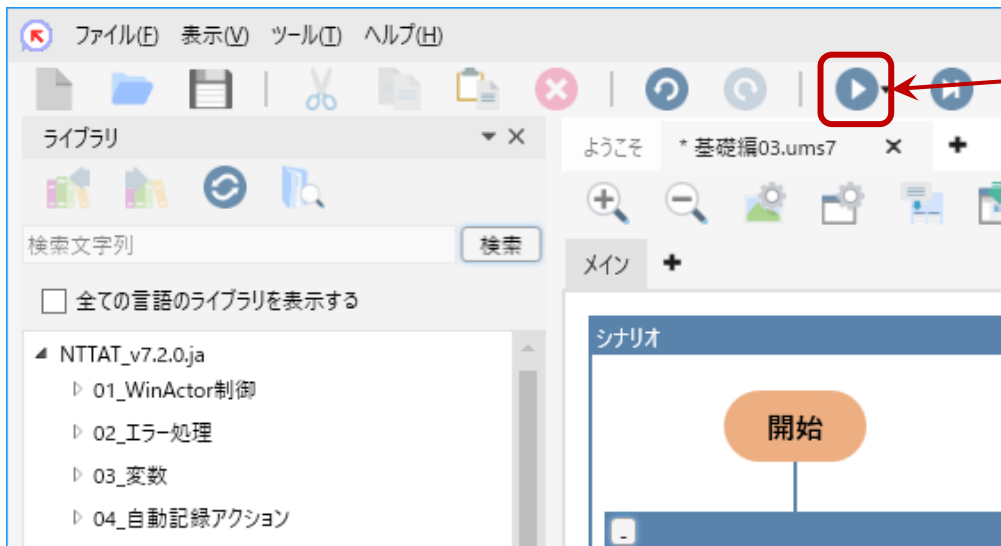
商品コード選択(3-4)

発注ボタンクリック(既存)

3

何度も同じ処理を繰り返し行う

3-7 シナリオの実行



1 <シナリオ実行> ボタンをクリックし、エラーなくシナリオが進むか確認します。

2 エラーが発生したら、エラーメッセージを確認しプロパティ等の設定を見直します。

3

何度も同じ処理を繰り返し行う

3-8 シナリオの保存



1 <ファイル> メニューをクリックします。

2 <名前を付けて保存> メニューをクリックします。

3 保存先を選択し、ファイル名を付けて保存します。



すでに保存されているファイルについては、<上書き保存> から保存することができます。



取得した文字や数字を加工する

導入

繰り返し処理をマスターし独り立ちできそうな宇院さんですが、WinActorで処理をする資料に対して事前に行う準備に思ったより時間がかかっていることに気づきました。



「…よく考えたら、元の資料に対して氏名のスペースを削ったり、数字のケタを合わせるために0で埋めたりしたり、意外と下処理に時間がかかっているな…。」



「そういった細かい処理もWinActorでできますよ。WinActorには文字や数字を編集する機能があり、これを使えば文字列の修正や削除、追加などができます。」



「それは便利です。今までメモ帳などで下処理していた処理もWinActorのできるのであればシナリオに組み込んでしまいたいです。」



「それでは実際に使ってみましょう。」



「実際には何をを使えば何ができるんですか？」

「数値の計算、文字の変換や連結、分割などができます。以下にいくつかの例をあげます。」

種別	ノード/ライブラリ名	内容	例
数値	四則演算	値と値で四則演算ができます。	$5 + 3 \rightarrow 8$
数値	不要な0を除去（整数）	桁合わせなどで頭に0がついている整数の0を除去します。	$000123 \rightarrow 123$
数値	カウントアップ	繰り返し処理でよく使用するノードです。対象の値に設定した整数を加算します。	$5 + 1 \rightarrow 6$
文字列	大文字化／小文字化	英字を大文字、または、小文字に変換します。	$ABC \Leftrightarrow abc$
文字列	トリミング	指定された文字列の先頭と末尾の空白を削除します。	(スペース) 123 (スペース) $\rightarrow 123$
文字列	文字列を前後に分割	「分割元文字列」に指定した文字列を、「区切文字列」で前後に分割します。	区切文字列 : ~ 2014/1/1~2014/3/31 ↓ 2014/1/1 と 2014/3/31
文字列	文字列の連結（2つ）	文字列1，文字列2 を連結して連結結果に格納します。	$ABC + DEF \rightarrow ABCDEF$

Point!





「例として＜四則演算＞のノードの設定を以下に示します。」

四則演算

名前 四則演算

コメント

計算結果 変数名を選択

変数名もしくは値を選択 + 変数名もしくは値を選択

☐ 整数として計算し、計算結果の小数点以下を切り捨てる

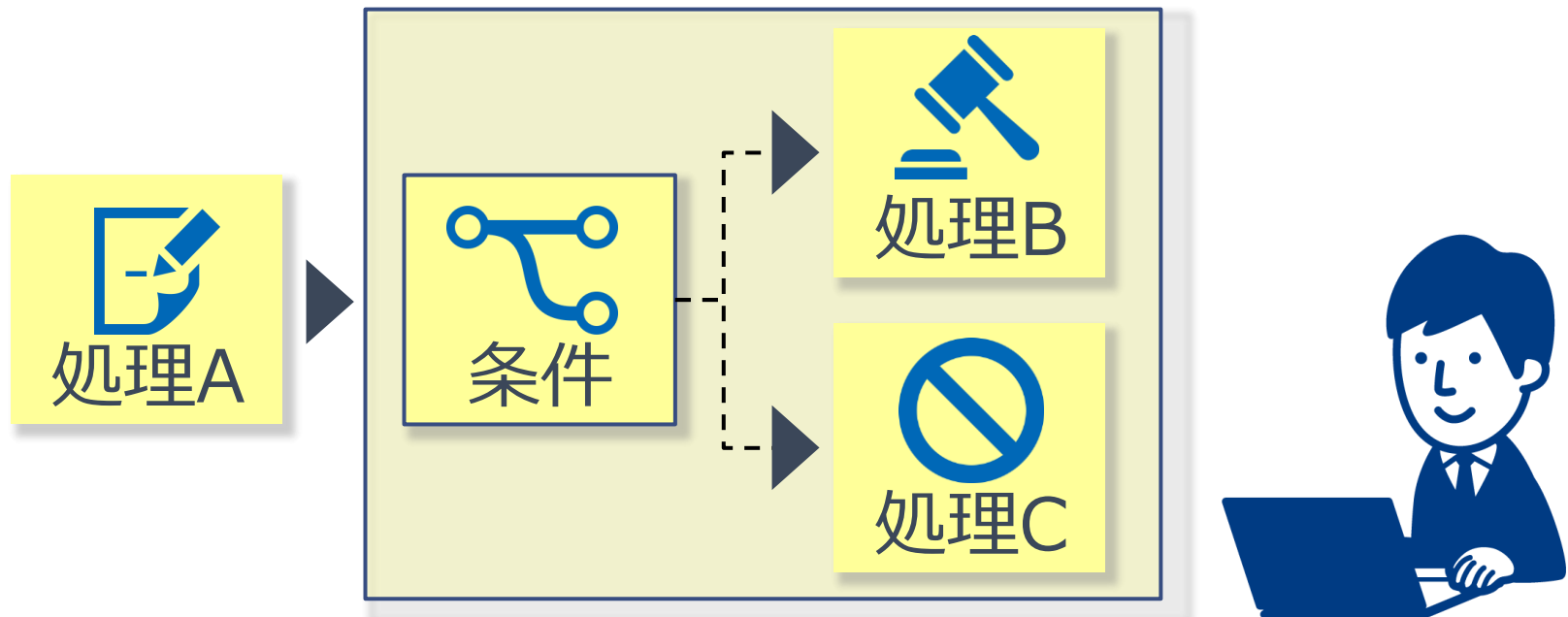
更新 元に戻す

1 <左のプルダウン>と<右のプルダウン>に、計算したい値を入力し、<演算子>に＋－×÷のどれかを設定します。

2 <更新> ボタンをクリックします。



「簡単ですね！」



条件によって処理内容を変更する

導入

繰り返し処理のシナリオ作成と文字列加工ができるようになった宇院さん。
今度は社内システムに入力する値によって処理を変えたいという依頼が来ました。



「前回教えてもらった繰り返し処理と文字列加工はすごい…100件でも1000件でも一気に下処理込みで実施できてしまうんだ。これは色々な作業に応用できそうだ！」



「それは良かったです。ただ、さきほど依頼を受けていた「決まった銀行であれば登録をする」という作業は繰り返し処理でできそうですか？」

?



「うーん…いえ、同じ作業の繰り返しだけではないので、今の私の知識では…」



「そうですね、そういった処理を条件分岐処理と言います。繰り返し処理と同じくらい重要な概念なので今回ぜひ覚えましょう。」

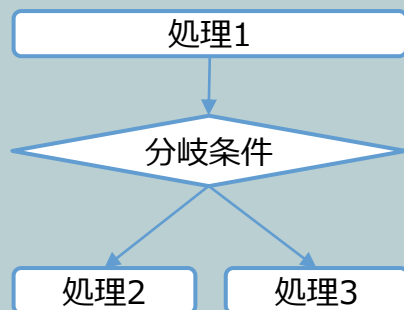


「そもそも条件分岐って何ですか…と聞きたいところですが、何らかの条件を付けて処理を分岐させること…ということですよね？」

「はい、その通りです！ 条件分岐とは○ならAを行い、×ならBを行うというように、条件合わせて実行される処理が異なります。」



Point!



■ 分岐条件
入力された値が「10」なら処理2を行う、「10」以外なら処理3を行う

「左図は条件分岐処理のイメージです。条件を満たす場合は2処理を行い、満たさない場合は3処理を行います。」

「繰り返し処理と同じく条件をどう設定するかがポイントになります。」

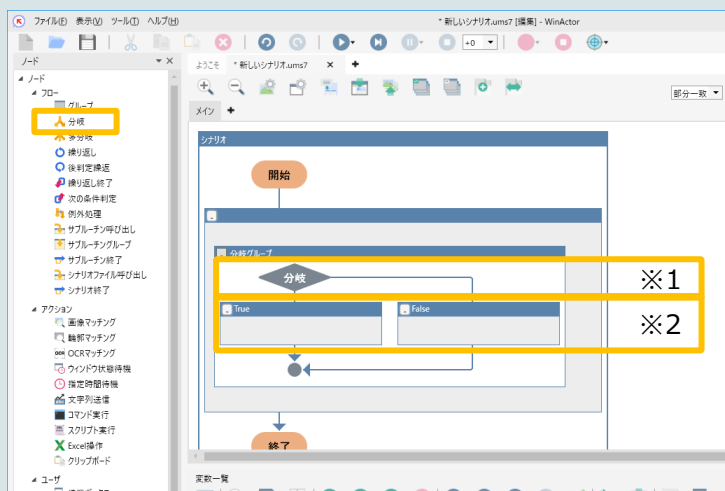


「実際のシナリオ作成では、どのようにすればいいですか？」

「画面左上のノードタブで表示される、＜分岐ノード＞を使用します」
「条件分岐処理は、ひし形の分岐条件(※1)と、条件を満たす(True)/満たさない(False)、それぞれの処理を設定する四角のエリア(※2)の二つで出来ています。」



Point!

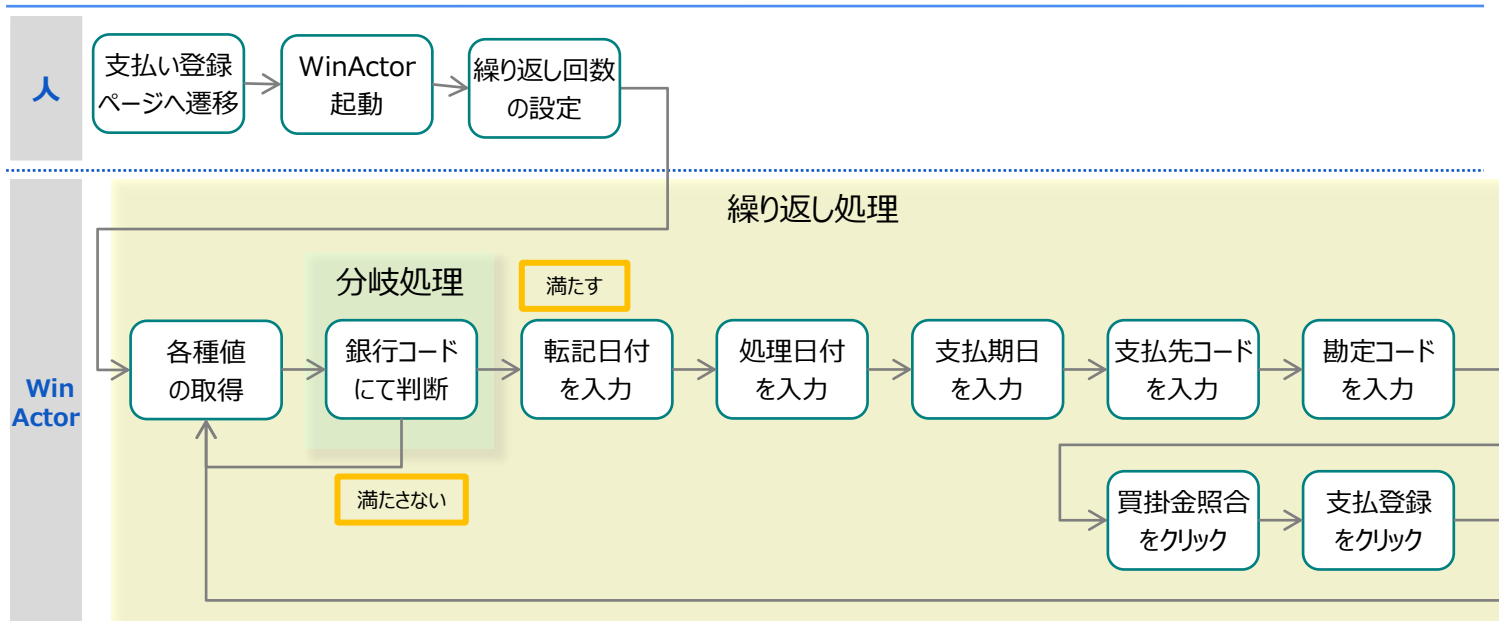


「条件と処理内容を設定することで、分岐処理ができます。条件はひし形をダブルクリックして表示される詳細設定画面で設定し、処理内容は他のシナリオと同じようにノードやライブラリをドラッグ&ドロップすることで処理を設定することができます。」



「今まで学んだ内容を応用して、支払い登録を行うシナリオを作ってみましょう。」

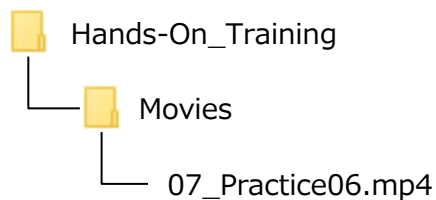
処理の流れ



「基礎編、実践編の総決算ですね！」



「それでは実際のWinActorの動きを見てみましょう。以下フォルダのファイル名をダブルクリックして再生してください。」



「それでは実際にシナリオを一緒に作っていきましょう！」

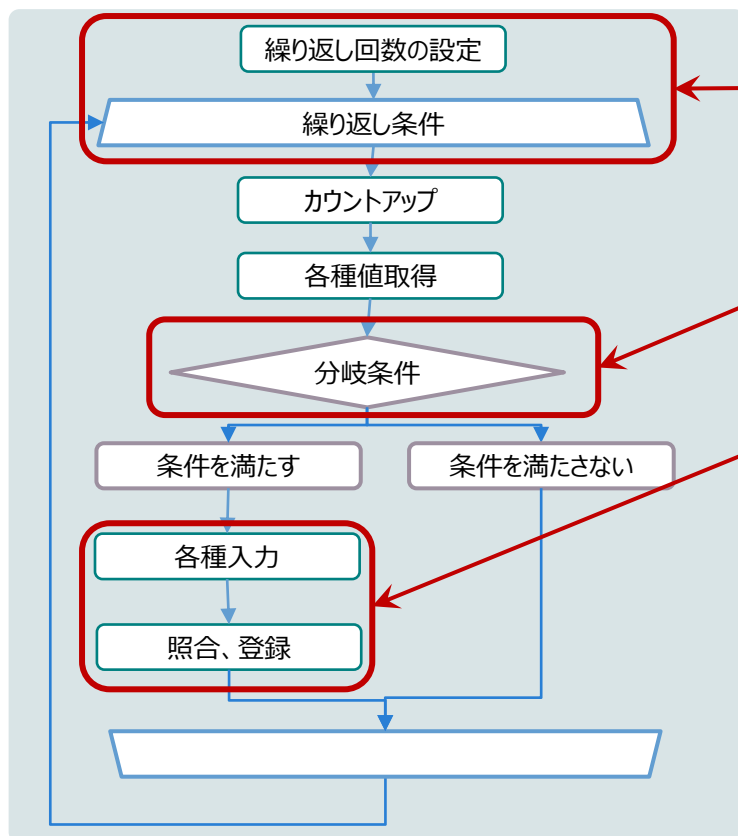
6

条件によって処理内容を変更する

準備

シナリオの流れ

1 今回作成するシナリオは以下のような流れになります。



2 前回と同様に台帳の行数を数えて繰り返しの数を設定します。

3 分岐条件に、＜銀行コードがNAT0001だったら＞を設定します。

4 分岐条件を満たす場合のみ、登録を行います。

6

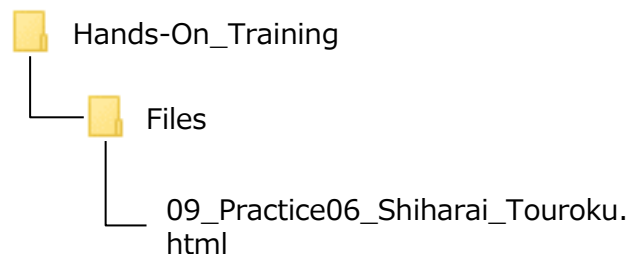
条件によって処理内容を変更する

準備

ログイン画面の表示

- 1 「09_Practice06_Shiharai_Touroku.html」を右クリックして、「プログラムから開く」から「Microsoft Edge」を選択します。

使用するファイル



デモシステムを起動するブラウザには
Edge または Chrome を使用してください。



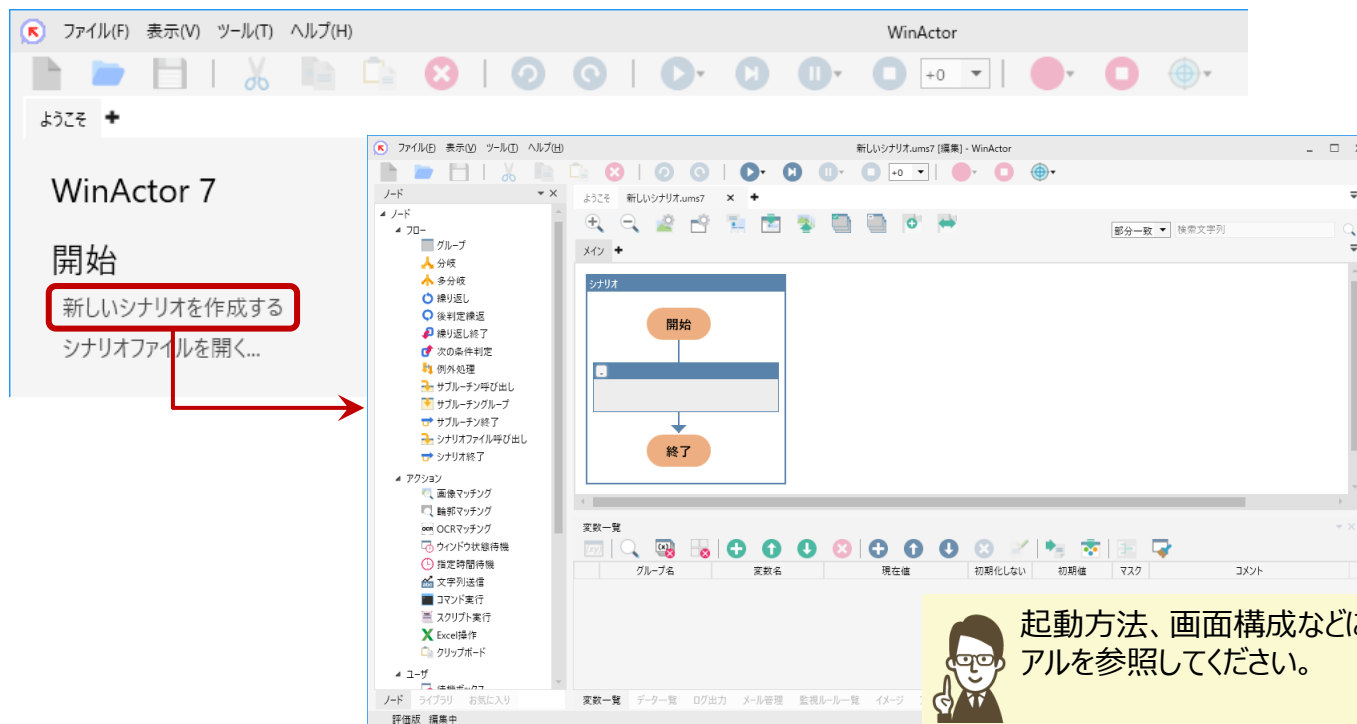
Chromeを使う人は「プログラムから開く」から
「Google Chrome」を選択してください。

6

条件によって処理内容を変更する

準備 WinActorの起動

- 1 スタートメニューなどからWinActorを起動します。
- 2 「新しいシナリオを作成する」をクリックすると、WinActorのメイン画面が表示されます。



起動方法、画面構成などについては操作マニュアルを参照してください。

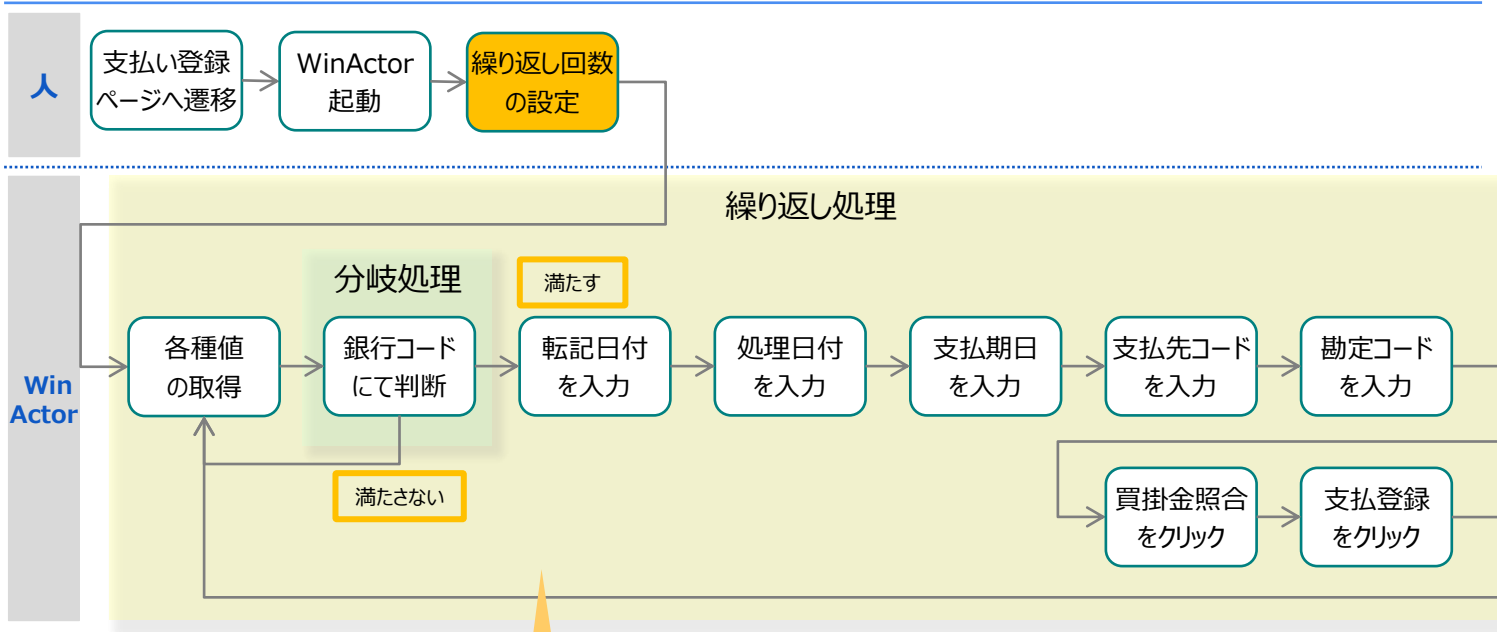
6

条件によって処理内容を変更する



「これから以下の業務についてシナリオを作成していきます。」

処理の流れ

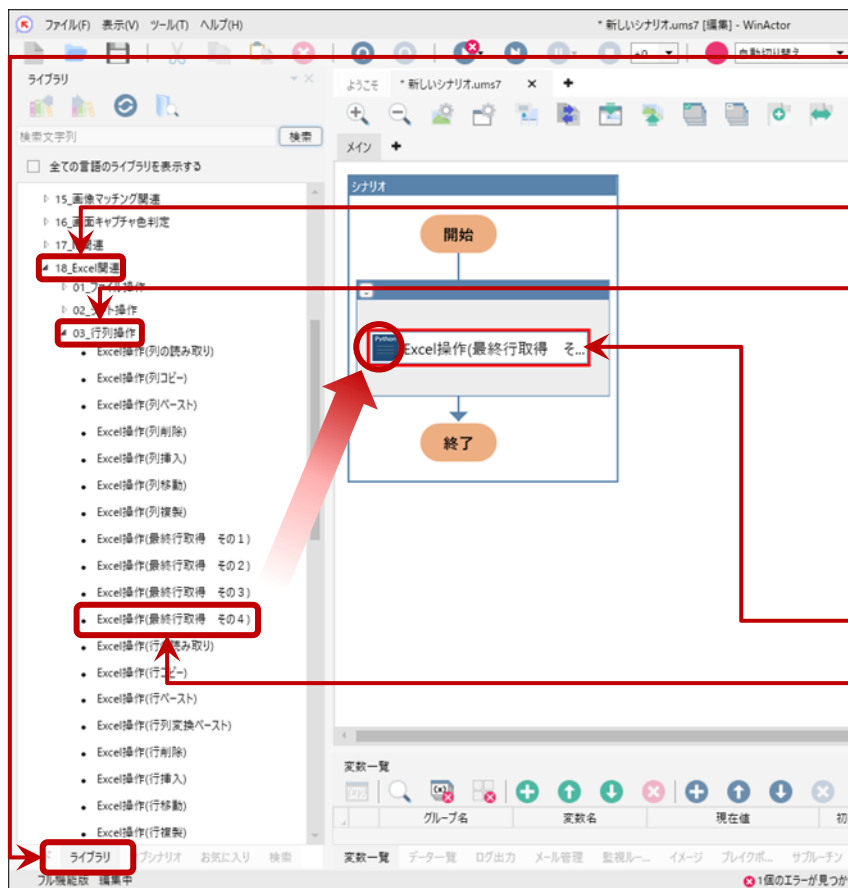


6-1 繰り返し回数の設定

6

条件によって処理内容を変更する

6-1 繰り返し回数の設定



1 <ライブラリ> をクリックします。

2 <18_Excel関連> をダブルクリックして、リストを展開します。

3 <03_行列操作> をダブルクリックして、リストを展開します。

4 <Excel操作(最終行取得 其の4)> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

5 配置したライブラリをダブルクリックします。(プロパティを表示)

最終行の行数を取得した後、不要な行を削除し繰り返し回数を設定します。
まずは台帳のすべての情報を取得したいので、最後の行まで取得してくれる<Excel操作(最終行取得 其の4)>というライブラリを活用します。



6

条件によって処理内容を変更する

6-1 繰り返し回数の設定

プロパティ Python実行

名前 Excel操作(最終行取得 その4)

コメント

設定 スクリプト 注釈 バージョン情報

[ファイル名] : 操作対象のファイルパスを指定してください。
相対パス、絶対パス、またはURLが利用可能です。

[シート名] : 操作対象のシート名を指定してください。
空の場合はアクティブなシートが選択されます。

[検索列] : 最終行を検索する列を指定してください。
例) A

[最終行] : 取得した最終行を格納する変数を指定してください。

ファイル名 値⇒ C:\temp¥10_Practice06_Shiharai_

シート名 値⇒ Sheet1

検索列 値⇒ C

最終行 最終行

更新 元に戻す

6

<ファイル名> に、
10_Practice06_Shiharai_Touroku_Daicho.xls
xをドラッグ & ドロップし、
<シート名> に、「値⇒ Sheet1」
<検索列> に、「値⇒ C」
<最終行> に、「最終行」
を入力します。

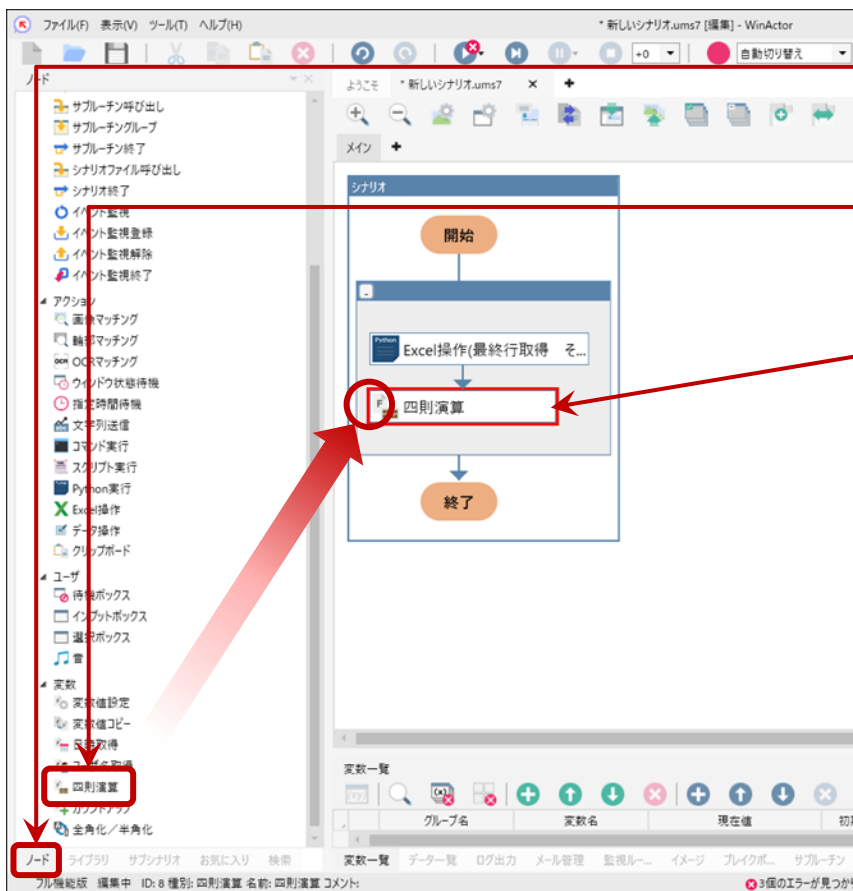
7

<更新> ボタンをクリックします。変数登録確認のダイア
ログは <はい> をクリックします。

6

条件によって処理内容を変更する

6-1 繰り返し回数の設定



8 <ノード> をクリックします。

9 <四則演算> ノードをドラッグし、シナリオ編集エリア上でドロップします。

10 配置したノードをダブルクリックします。(プロパティを表示)

6

条件によって処理内容を変更する

6-1 繰り返し回数の設定

11 <計算結果>に「繰り返し回数」を入力します。

12 左の<変数名もしくは値を選択>のプルダウンから、「最終行」を選択、中央の<演算子>のプルダウンから、「－（マイナス）」を選択、右の<変数名もしくは値を選択>のプルダウンから、「値⇒4」を入力します。

13 <更新> ボタンをクリックします。変数登録確認のダイアログは<はい>をクリックします。

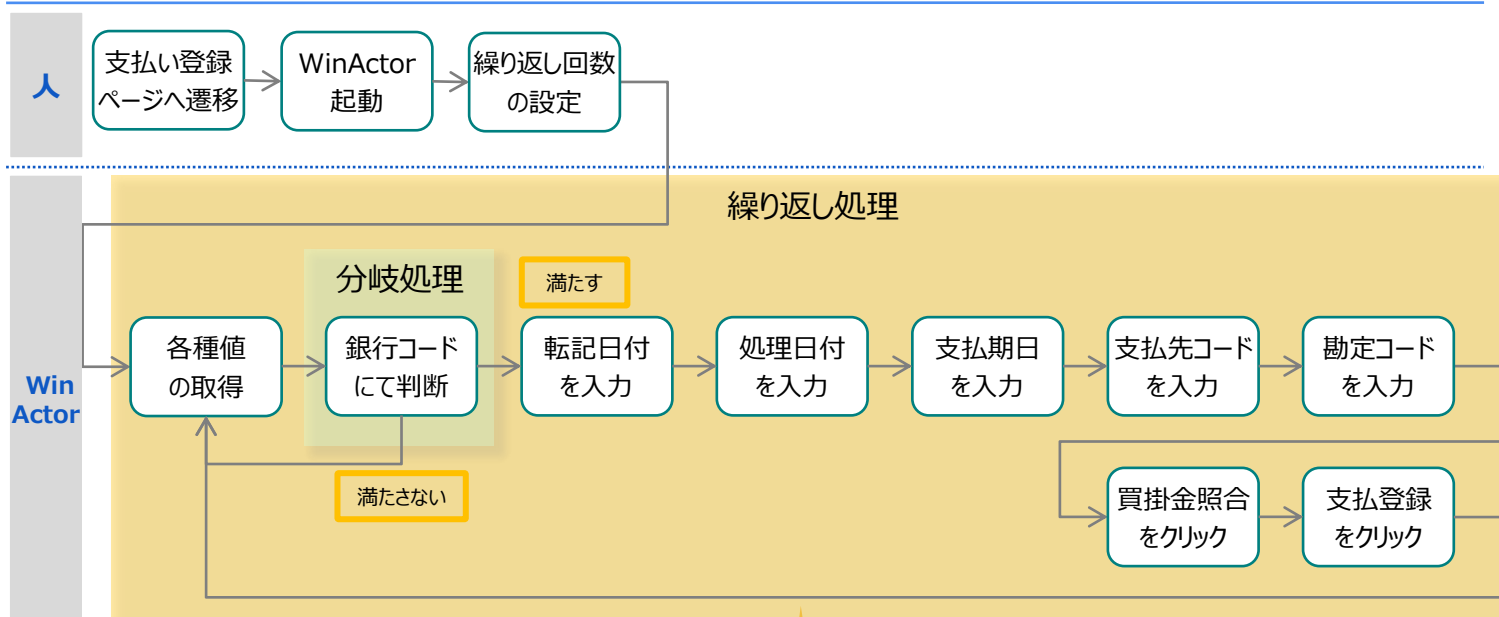
6

条件によって処理内容を変更する



「これから以下の業務についてシナリオを作成していきます。」

処理の流れ



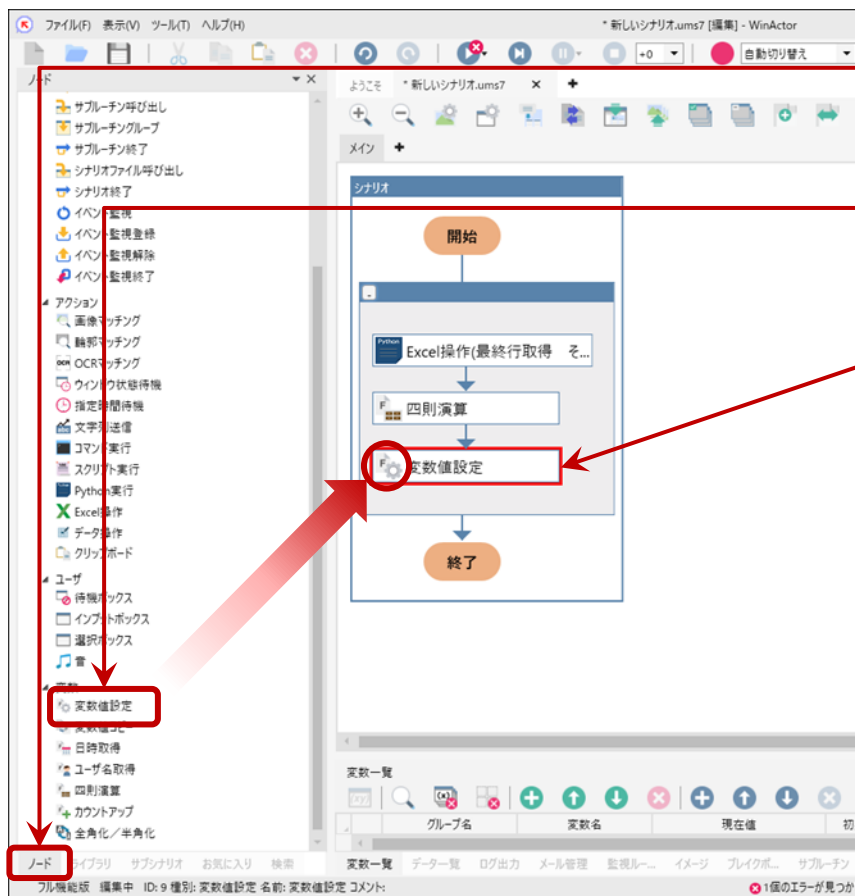
6-2 台帳の入力開始行設定

6-3 繰り返し処理の設定

6

条件によって処理内容を変更する

6-2 台帳の入力開始行設定



1 <ノード> をクリックします。

2 <変数値設定> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したノードをダブルクリックします。(プロパティを表示)

6

条件によって処理内容を変更する

6-2 台帳の入力開始行設定

プロパティ

変数値設定

名前 変数値設定

コメント

変数名 カウンター

設定値 4

更新

元に戻す

4 <変数名> に、「カウンター」を入力します。

5 <設定値> に、「4」を入力します。

6 <更新> ボタンをクリックします。変数登録確認のダイアログは<はい> をクリックします。

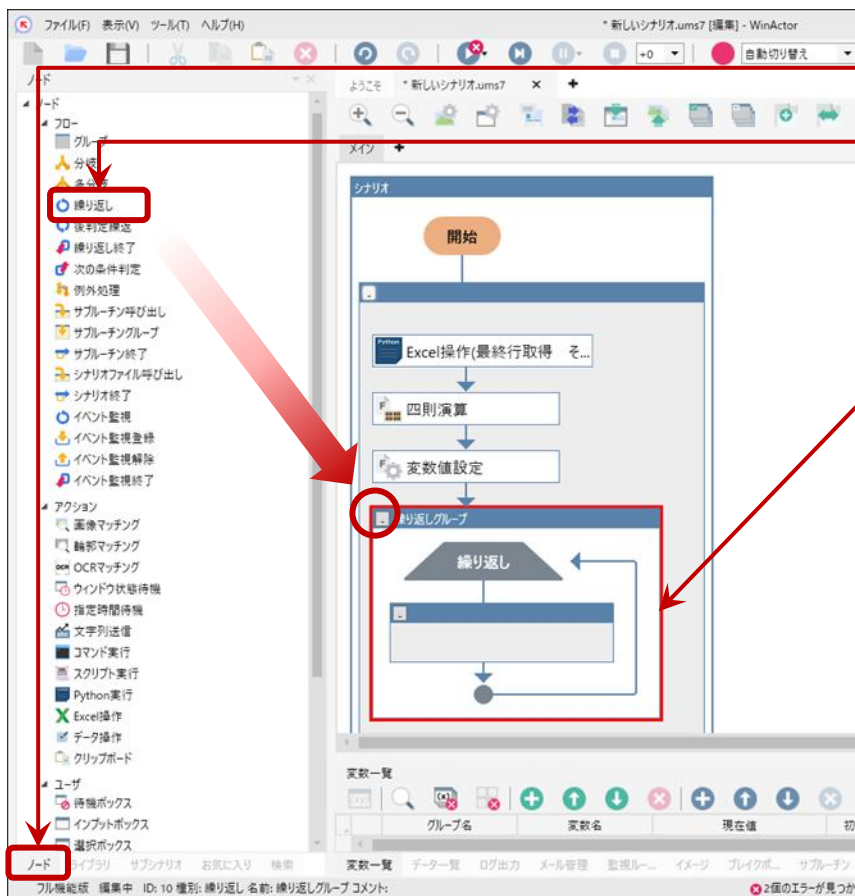


カウンター変数にあらかじめ4を設定しておくのは、台帳の5行目から入力を行いたいからです。繰り返し処理の中の初めにカウンターに+ 1 を行うため 4 と設定しています。

6

条件によって処理内容を変更する

6-3 繰り返し処理の設定



1 <ノード> をクリックします。

2 <繰り返し> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したノードをダブルクリックします。(プロパティを表示)

6

条件によって処理内容を変更する

6-3 繰り返し処理の設定

繰り返し

名前 繰り返しグループ

コメント

☐ 条件式 ☒ 回数

条件式設定

繰り返し回数

☐ 範囲 変数名もしくは値を選択 から 変数名もしくは値を選択

☐ データ数 データファイル名 変数名もしくは値を選択

☐ データ数 (DB連携) データソース名 変数名もしくは値を選択

ユーザ名 変数名もしくは値を選択

パスワード 変数名もしくは値を選択

テーブル名 変数名もしくは値を選択

カウンタ 変数名を選択 (省略可)

更新 元に戻す

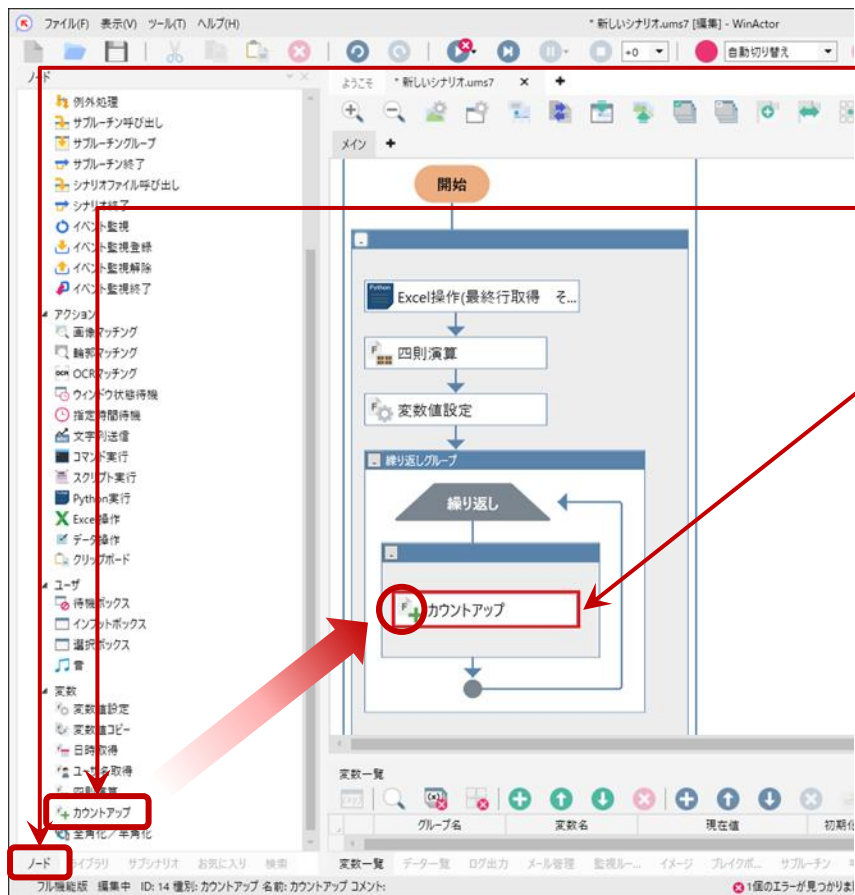
4 <回数> のラジオボタンをクリックし、「繰り返し回数」を選択します。

5 <更新> ボタンをクリックします。

6

条件によって処理内容を変更する

6-3 繰り返し処理の設定



6 <ノード> をクリックします。

7 <カウントアップ> ノードをドラッグし、シナリオ編集エリア上でドロップします。

8 配置したノードをダブルクリックします。(プロパティを表示)

6

条件によって処理内容を変更する

6-3

繰り返し処理の設定

カウントアップ

名前 カウントアップ

コメント

計算結果 カウンター

加算値 1

更新 元に戻す

9 <計算結果> に、「カウンター」を選択します。

10 <加算値> に、「1」を選択します。

11 <更新> ボタンをクリックします。

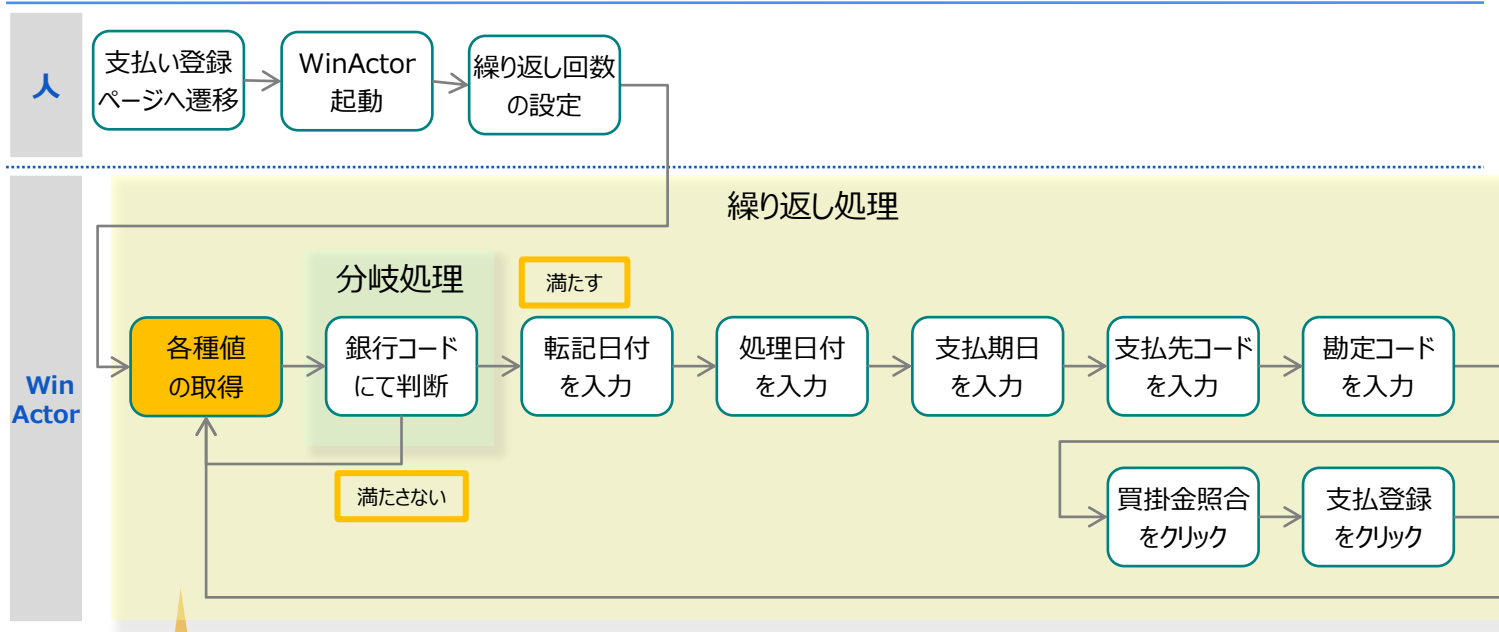
6

条件によって処理内容を変更する



「これから以下の業務についてシナリオを作成していきます。」

処理の流れ



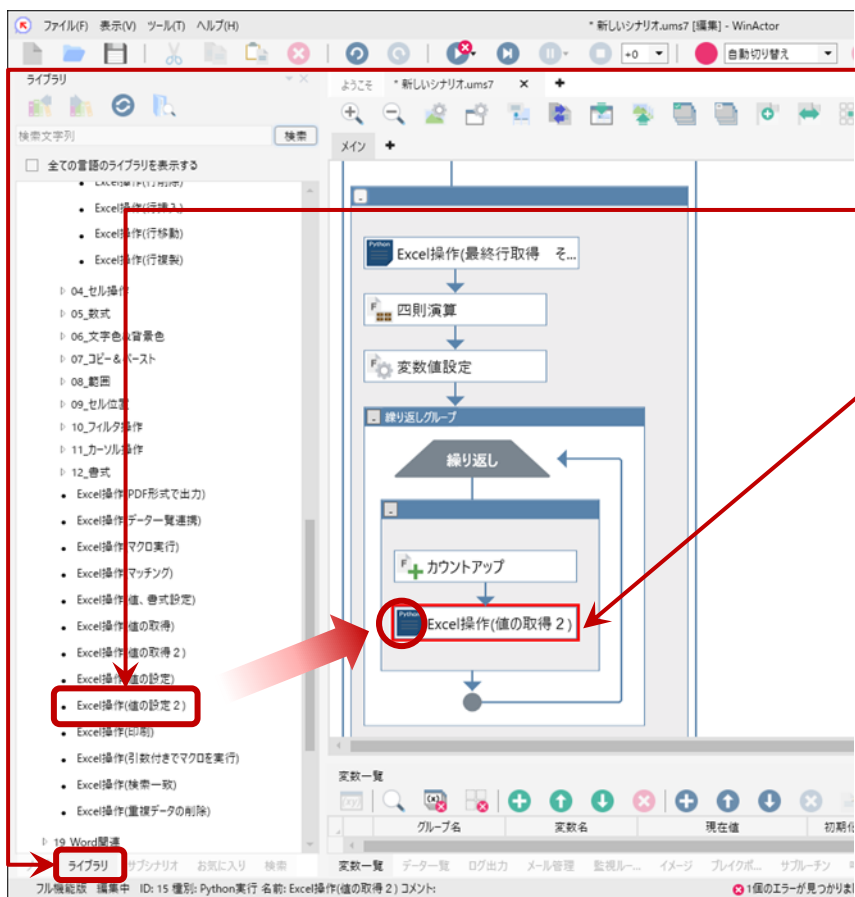
6-4

値取得設定

6

条件によって処理内容を変更する

6-4 値取得設定



1 <ライブラリ> をクリックします。

2 <Excel操作(値の取得2)> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したライブラリをダブルクリックします。(プロパティを表示)

6

条件によって処理内容を変更する

6-4

値取得設定

プロパティ Python実行

名前 Excel操作(値の取得 2)

コメント

設定 スクリプト 注釈 バージョン情報

表示設定が日付や時刻のセルから表示文字列を取得したい場面などで選択してください。

[格納先変数]
取得した値を格納する変数を指定してください。

ファイル名 値⇒ C:\temp¥10_Practice06-Shiharai_

シート名 値⇒ Sheet1

セル(行) カウンター

セル(列) 値⇒ C

取得方法 value

格納先変数 支払期日

更新 元に戻す

4

<ファイル名> に、
10_Practice06_Shiharai_Touroku_Daicho.xlsx をドラッグ & ドロップし、
<シート名> に、「値⇒ Sheet1」
<セル(行)> に、「カウンター」
<セル(列)> に、「値⇒ C」
<取得方法> に、「value」
<格納先変数> に、「支払期日」
を入力します。

5

<更新> ボタンをクリックします。変数登録確認のダイアログは <はい> をクリックします。

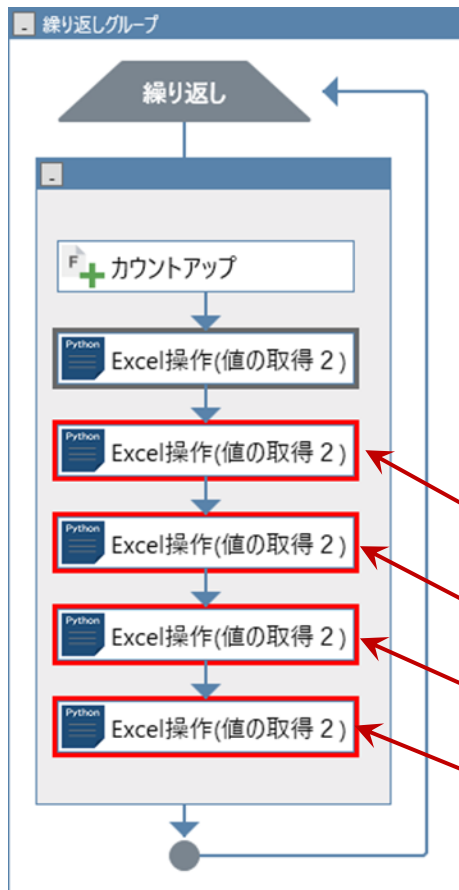
この設定で、<セル(行)> に、「カウンター」を設定しているのがポイントです。



「カウンター」は繰り返しを行うごとに 1 ずつカウントアップされていくため、繰り返される度に行番号が 1 ずつ増えていきます。

6-4

値取得設定



前述した支払期日を取得する設定のように、銀行コード・支払先コード・勘定コード・支払金額についても設定します。
<セル(列)>と<格納先変数>については下表を参考に設定ください。

6

<ファイル名>、<シート名>、<セル(行)>、<取得方法>については、前述同様に、
<ファイル名>に、
10_Practice06_Shiharai_Touroku_Daicho.xlsx を
ドラッグ&ドロップし、
<シート名>に、「値⇒ Sheet1」
<セル(行)>に、「カウンター」
<取得方法>に「value」を入力します。

セル(列)	格納先変数
D	銀行コード
E	支払先コード
F	勘定コード
G	支払金額

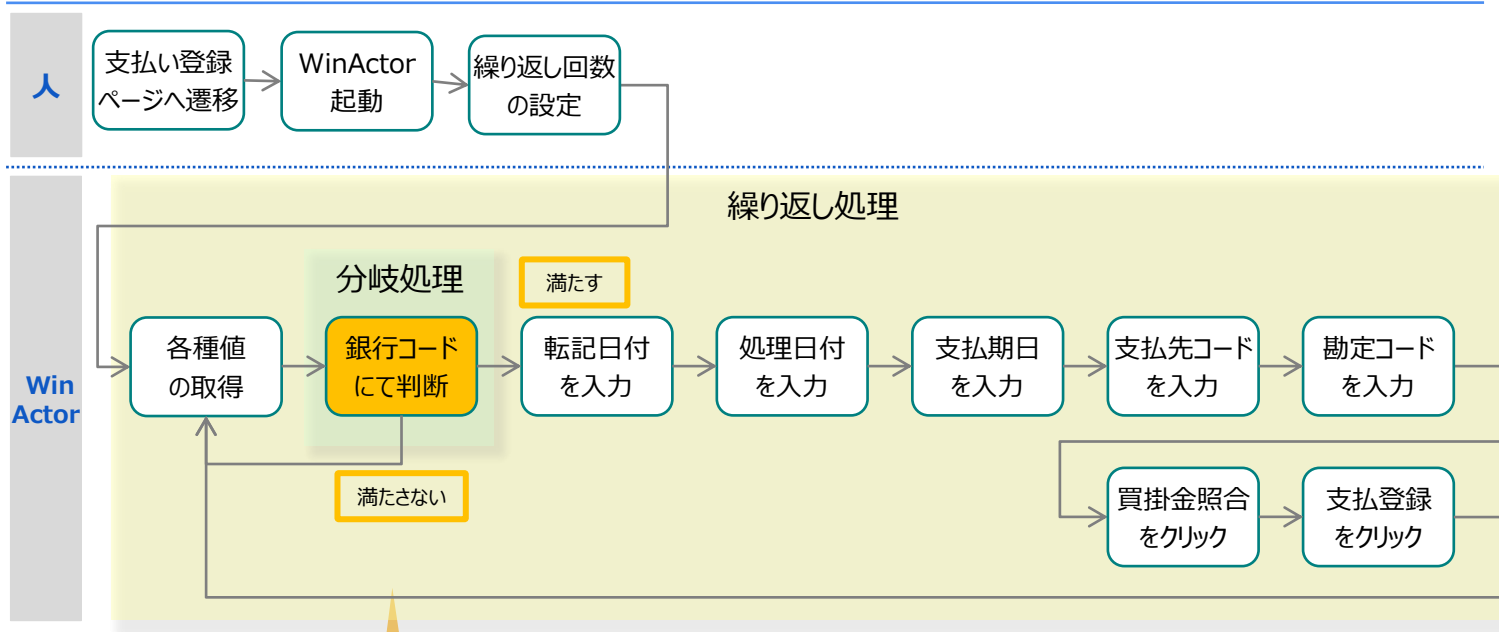
6

条件によって処理内容を変更する



「これから以下の業務についてシナリオを作成していきます。」

処理の流れ



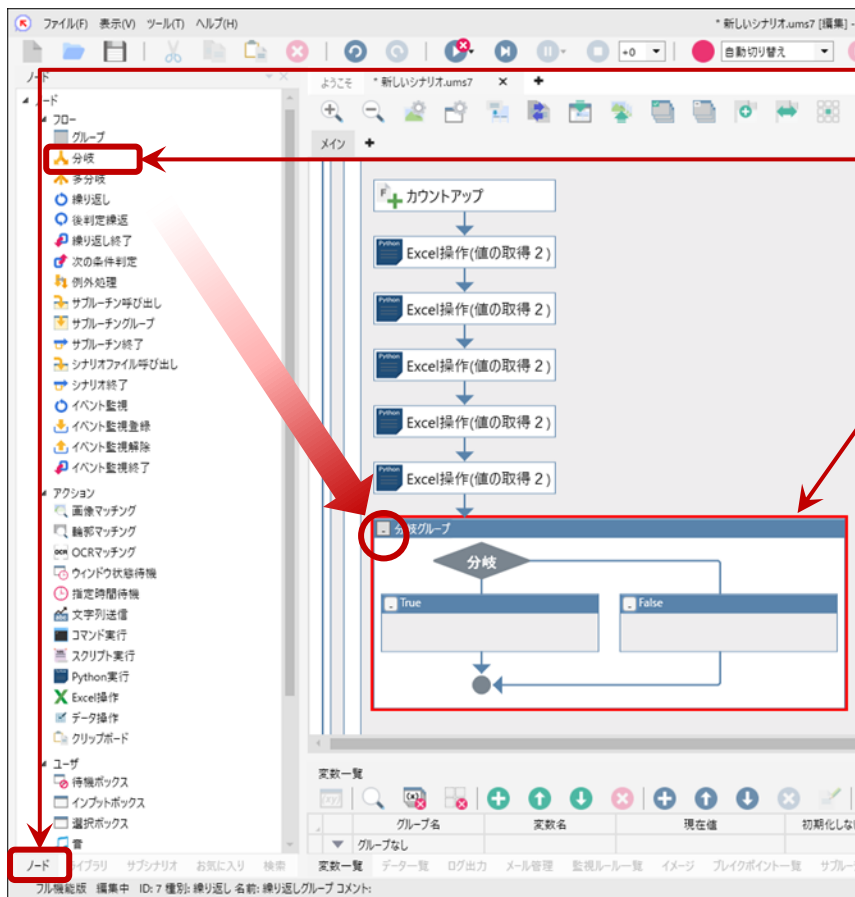
6-5

分岐処理設定

6

条件によって処理内容を変更する

6-5 分岐処理設定



1 <ノード> をクリックします。

2 <分岐> ノードをドラッグし、シナリオ編集エリア上でドロップします。

3 配置したノードをダブルクリックします。(プロパティを表示)

6

条件によって処理内容を変更する

6-5 分岐処理設定

分岐

名前 分岐グループ

コメント

判定	分岐名	条件式
真	True	[] 等しい []
偽	False	

条件式設定

4 <条件式設定> を、クリックします。

条件式

条件

変数名もしくは値を選択

等しい

変数名もしくは値を選択

x

5 <条件> から、「AND」を選択します。

条件式

AND

+

6 <+> を、クリックします。

6

条件によって処理内容を変更する

6-5 分岐処理設定

条件式

AND + ×

条件 銀行コード 等しい 値 ⇒ NAT0001 ×

更新 元に戻す

7

＜条件＞に、「条件」「銀行コード」「等しい」「値⇒NAT0001」を設定します。

8

＜更新＞ボタンをクリックします。

9

＜更新＞ボタンをクリックします。

分岐

名前 分岐グループ

コメント

判定	分岐名	条件式	
真	True	([銀行コード] 等しい)	条件式設定
偽	False		

更新 元に戻す

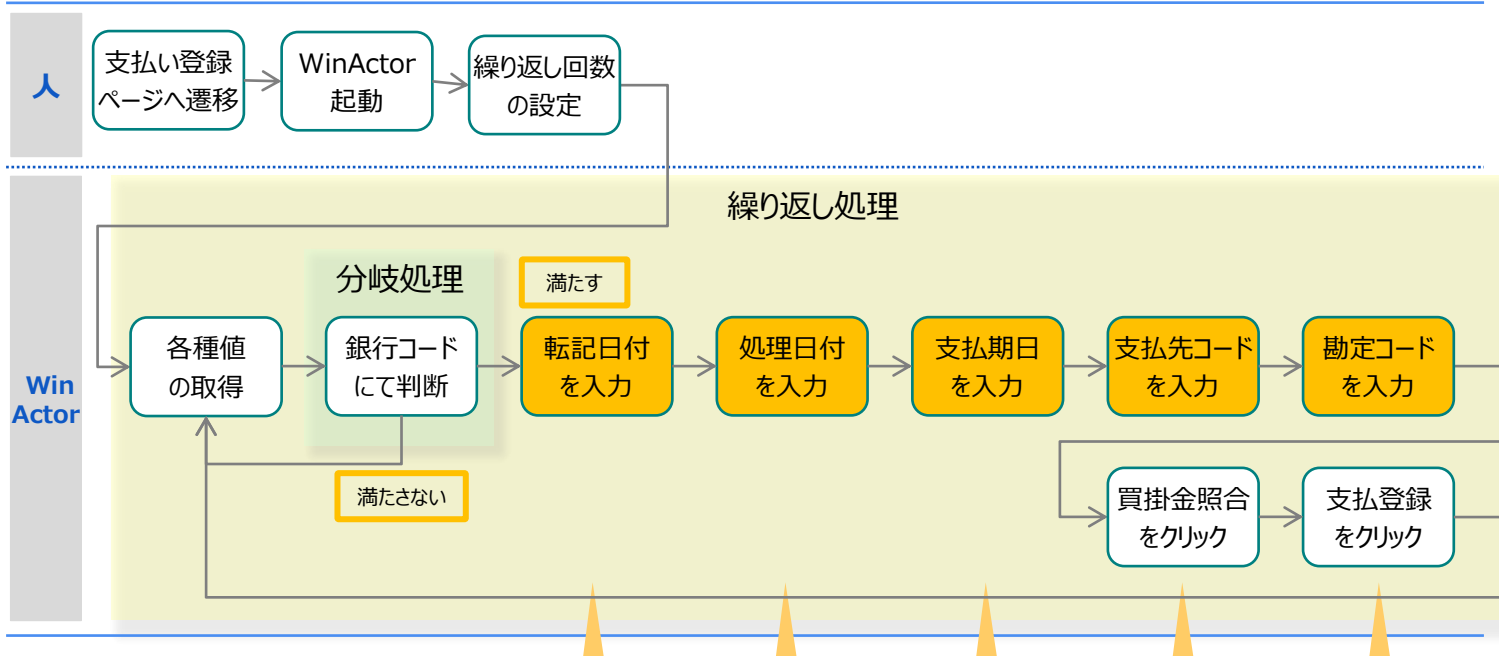
6

条件によって処理内容を変更する



「これから以下の業務についてシナリオを作成していきます。」

処理の流れ



6-6

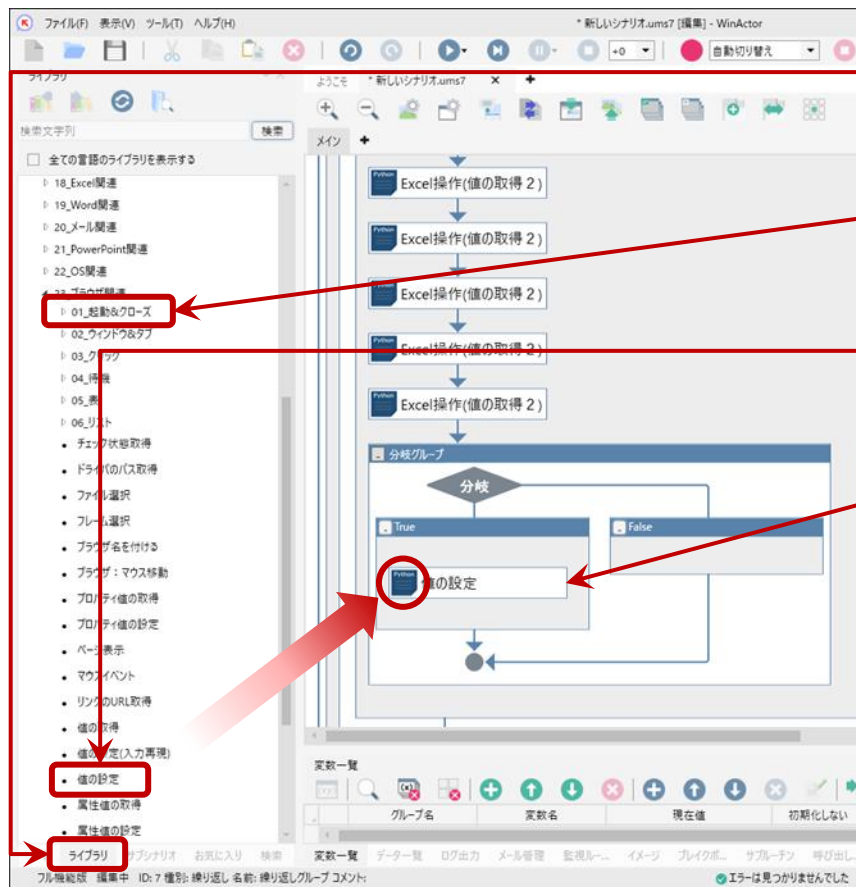
値入力設定

6

条件によって処理内容を変更する

6-6

値入力設定



1 <ライブラリ> をクリックします。

2 <23_ブラウザ関連> をダブルクリックして、リストを展開します。

3 <値の設定> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

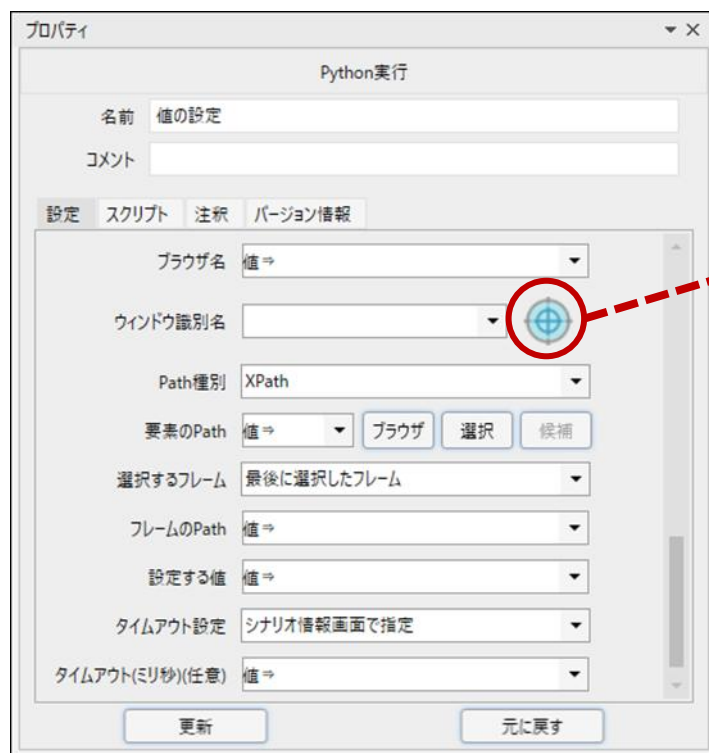
6

条件によって処理内容を変更する

6-6 値入力設定

5 <ウィンドウ識別名>の青いスコープボタンをクリックします。

6 操作したい画面上にマウスカーソルを移動し、クリックします。



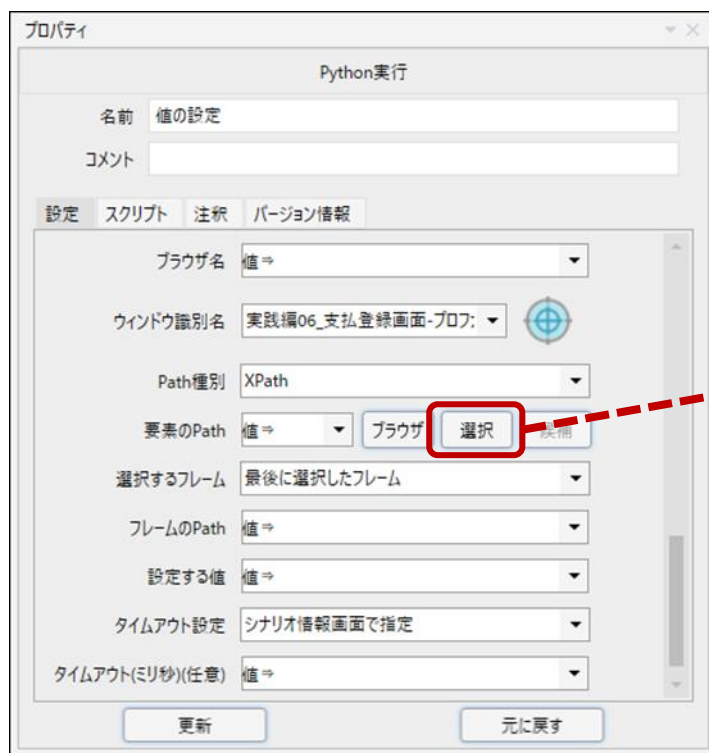
6

条件によって処理内容を変更する

6-6 値入力設定

7 <要素のPath> の<選択> ボタンをクリックします。

8 <転記日付> テキストボックスにマウスカーソルを移動し、色が変わったら、クリックします。



6

条件によって処理内容を変更する

6-6

値入力設定

プロパティ Python実行

名前 値の設定

コメント

設定 スクリプト 注釈 バージョン情報

ブラウザ名 値⇒

ウィンドウ識別名 実践編06_支払登録画面-プロフ:

Path種別 XPath

要素のPath 値⇒ //*[④ ブラウザ 選択 候補

選択するフレーム トップフレーム

フレームのPath 値⇒

設定する値 値⇒ 2019/12/20

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値⇒

更新 元に戻す

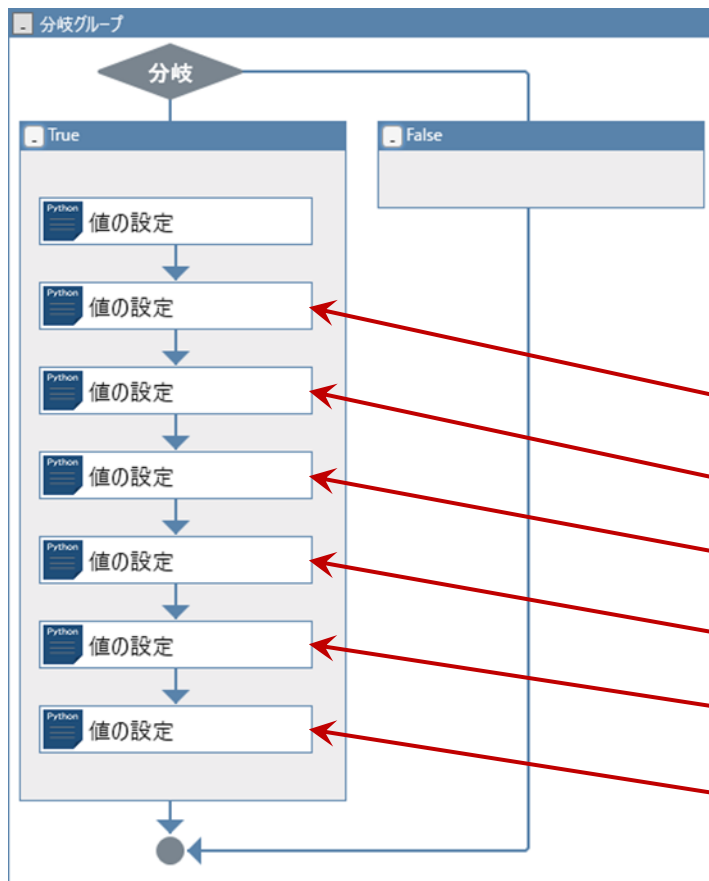
9

＜設定する値＞に、「値⇒ 2019/12/20」を入力します。

10

＜更新＞ボタンをクリックします。

6-6 値入力設定



11

前述した転記日付を入力する設定のように、処理日付、支払期日、銀行コード、支配先コード、勘定コード、支払金額についても設定します。
 <XPath>と<設定する値>については下表を参考に設定ください。

XPath	設定する値
処理日付入力ボックス	値⇒ 2019/12/26
支払期日入力ボックス	支払期日
銀行コード入力ボックス	銀行コード
支払先コード入力ボックス	支払先コード
勘定コード入力ボックス	勘定コード
支払金額入力ボックス	支払金額

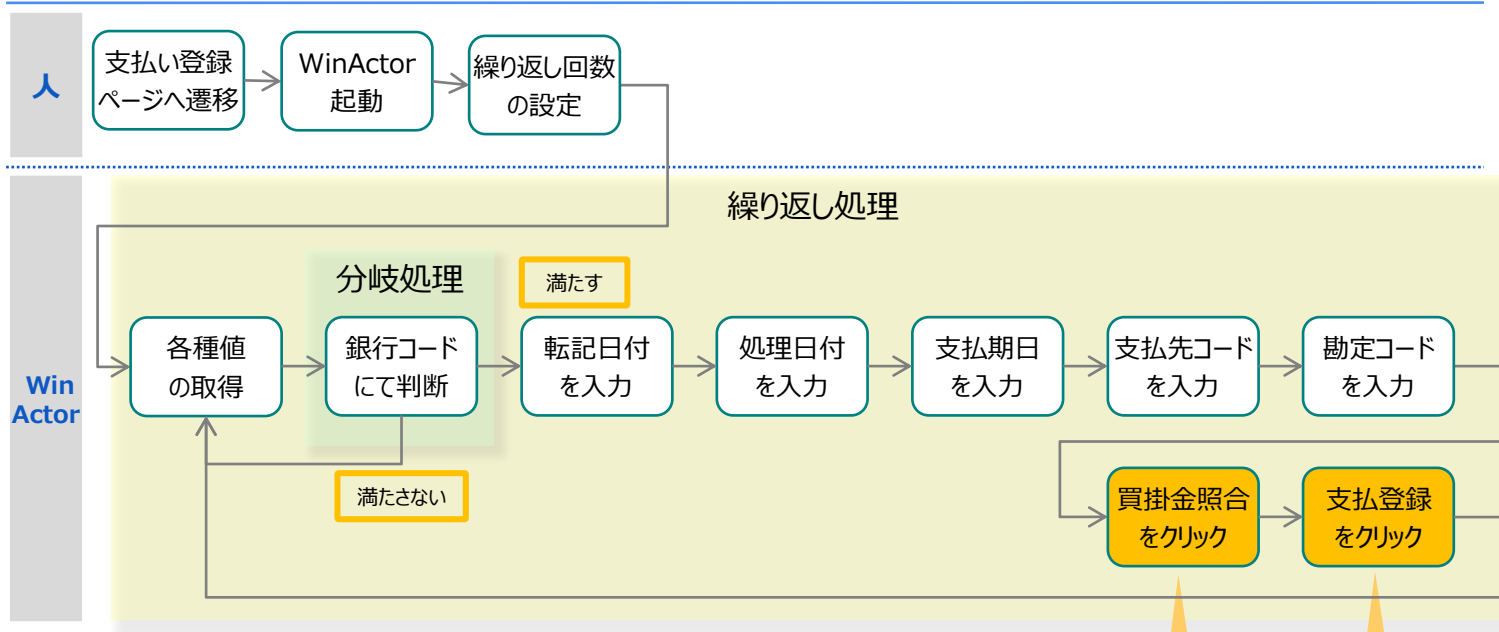
6

条件によって処理内容を変更する



「これから以下の業務についてシナリオを作成していきます。」

処理の流れ



6-7

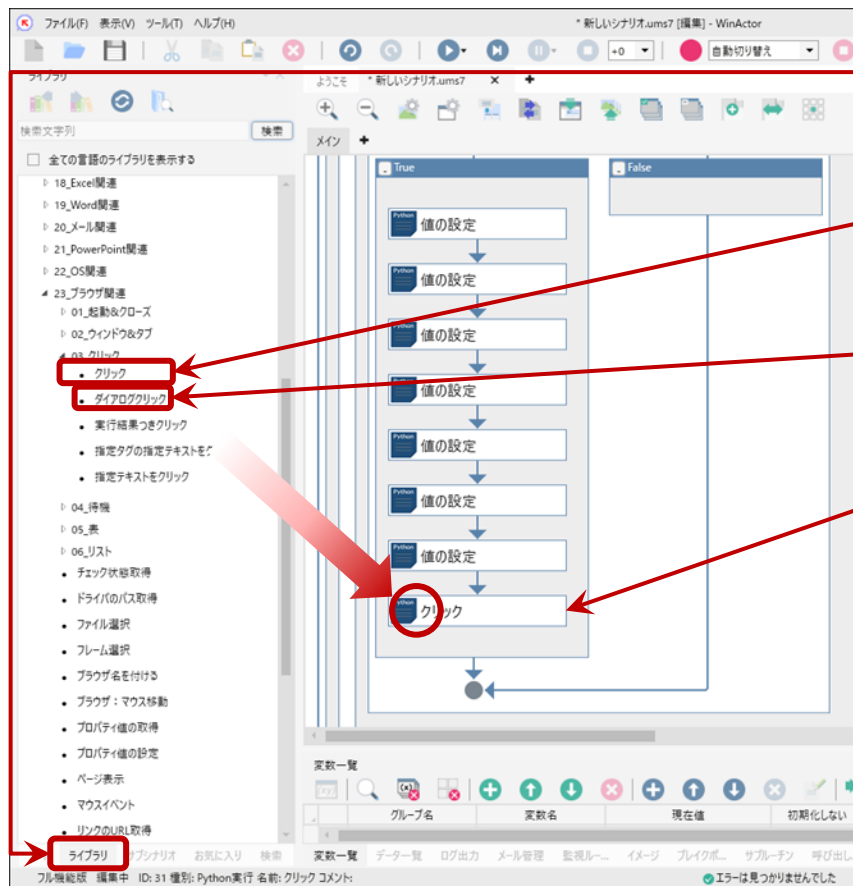
登録設定

6

条件によって処理内容を変更する

6-7

登録設定



1 <ライブラリ> をクリックします。

2 <03_クリック> をダブルクリックして、リストを展開します。

3 <クリック> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

4 配置したライブラリをダブルクリックします。(プロパティを表示)

6

条件によって処理内容を変更する

6-7

登録設定

5 <ウィンドウ識別名>の青いスコープボタンをクリックします。

6 操作したい画面上にマウスカーソルを移動し、クリックします。

プロパティ Python実行

名前 クリック

コメント

設定 スクリプト 注釈 バージョン情報

未設定の場合、10,000ミリ秒となります。

ブラウザ名 値→

ウィンドウ識別名 [スコープボタン]

Path種別 XPath

要素のPath 値→ ブラウザ 選択 候補

選択するフレーム 最後に選択したフレーム

フレームのPath 値→

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値→

更新 元に戻す

WinActorシナリオ デモシステム

メニュー 発注登録 発注一覧 入庫登録 支払登録

支払登録

帳記日付 2019/11/29 処理日付 2019/11/29

支払期日 2019/11/30

銀行コード NAT0001 銀行名

支払先コード 987654 支払先名

勘定コード 100990 勘定科目名

支払金額 8765000

保存 買掛金照合 支払登録

6

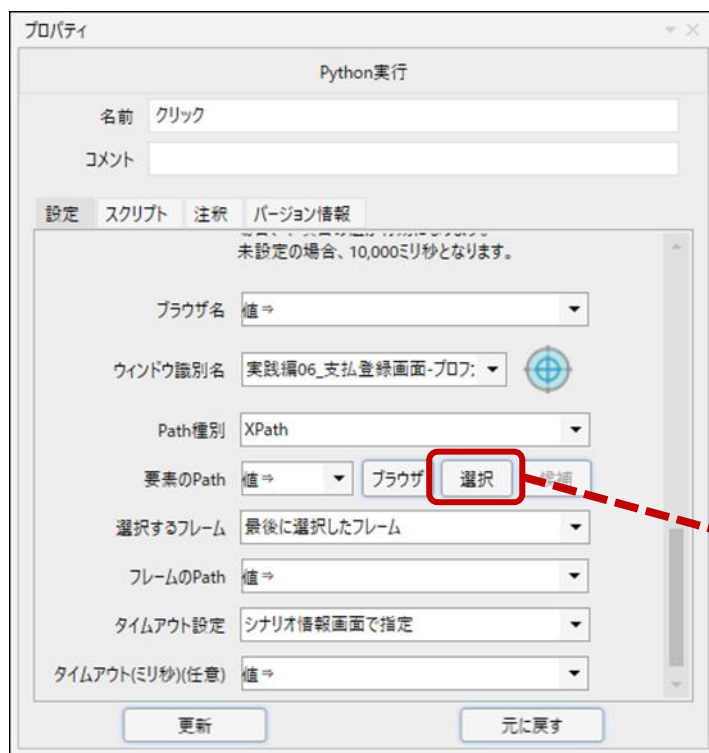
条件によって処理内容を変更する

6-7

登録設定

7 <要素のPath> の<選択> ボタンをクリックします。

8 <買掛金照合> ボタンにマウスカーソルを移動し、色が変わったら、クリックします。



6

条件によって処理内容を変更する

6-7

登録設定

9 <更新> ボタンをクリックします。

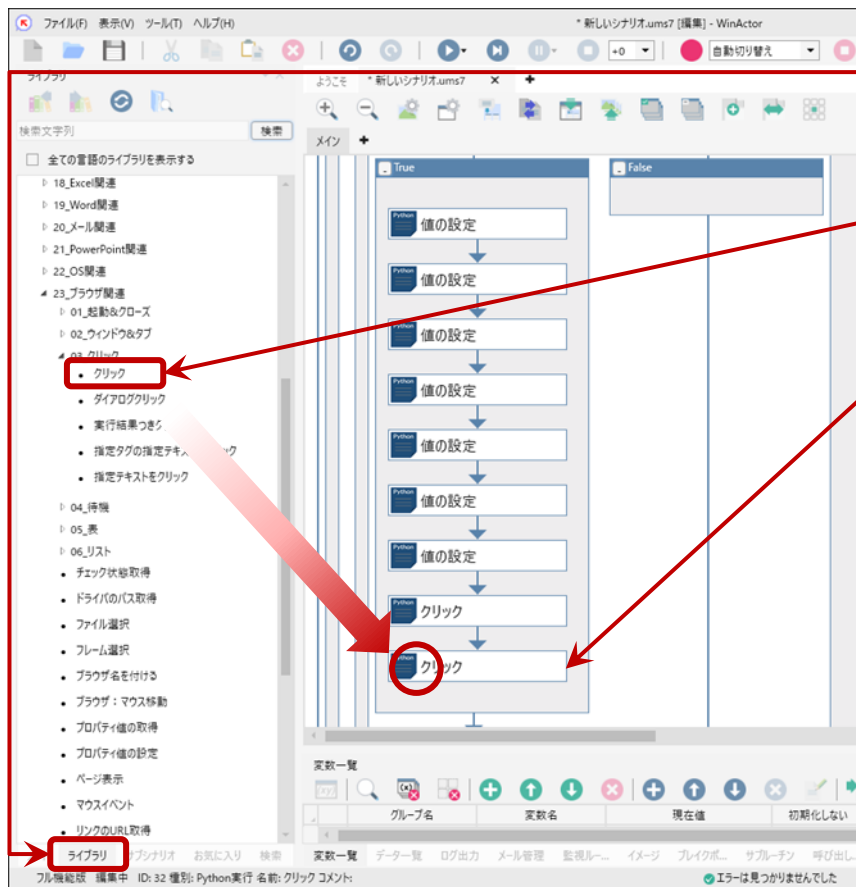
The screenshot shows a 'Python実行' (Python Execution) configuration window. The '名前' (Name) field is set to 'クリック' (Click). The 'コメント' (Comment) field is empty. The '設定' (Settings) tab is selected, showing various configuration options. The 'ブラウザ名' (Browser Name) is set to '値' (Value). The 'ウィンドウ識別名' (Window Identification Name) is set to '実践編06_支払登録画面-プロフ:'. The 'Path種別' (Path Type) is set to 'XPath'. The '要素のPath' (Element Path) is set to '//*[text()=値]'. The '選択するフレーム' (Select Frame) is set to 'トップフレーム'. The 'フレームのPath' (Frame Path) is set to '値'. The 'タイムアウト設定' (Timeout Setting) is set to 'シナリオ情報画面で指定'. The 'タイムアウト(ミリ秒)(任意)' (Timeout (ms) (Optional)) is set to '値'. The '更新' (Update) button at the bottom is highlighted with a red rectangle.

6

条件によって処理内容を変更する

6-7

登録設定



10 <ライブラリ> をクリックします。

11 <クリック> ライブラリをドラッグし、シナリオ編集エリア上でドロップします。

12 配置したライブラリをダブルクリックします。(プロパティを表示)

6

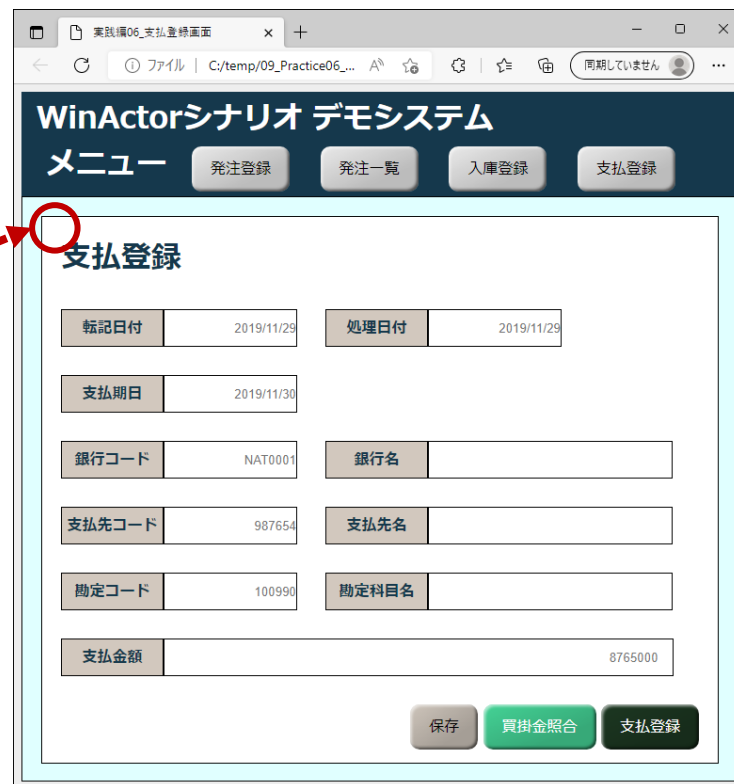
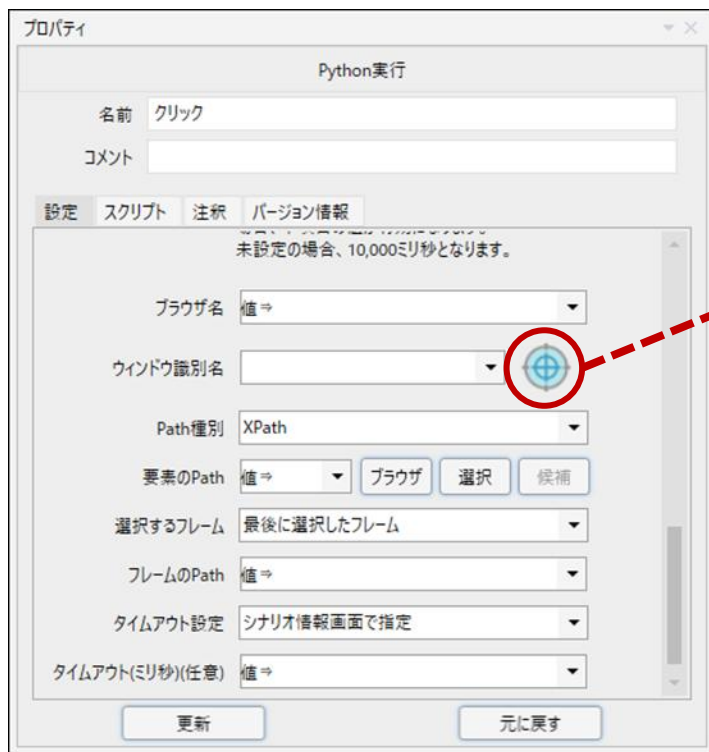
条件によって処理内容を変更する

6-7

登録設定

13 <ウィンドウ識別名>の青いスコープボタンをクリックします。

14 操作したい画面上にマウスカーソルを移動し、クリックします。



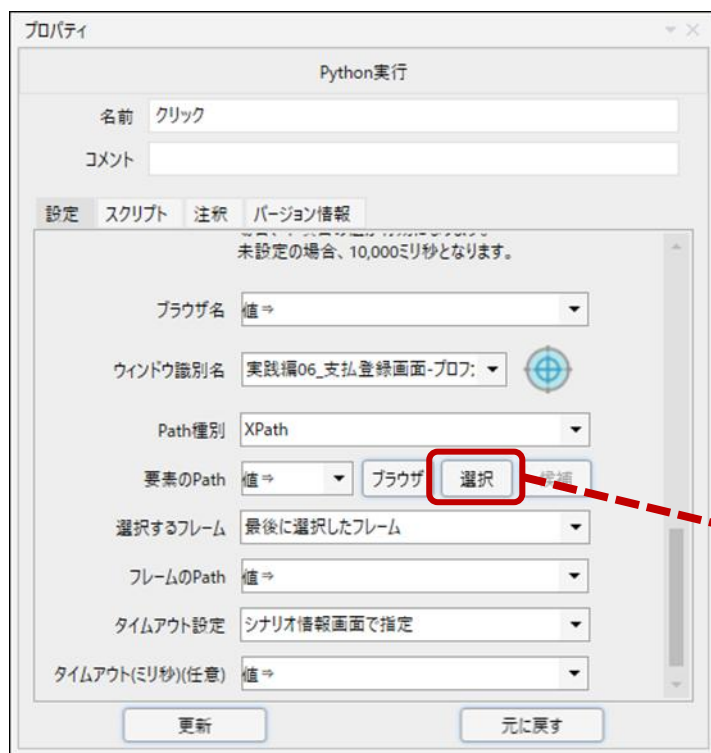
6

条件によって処理内容を変更する

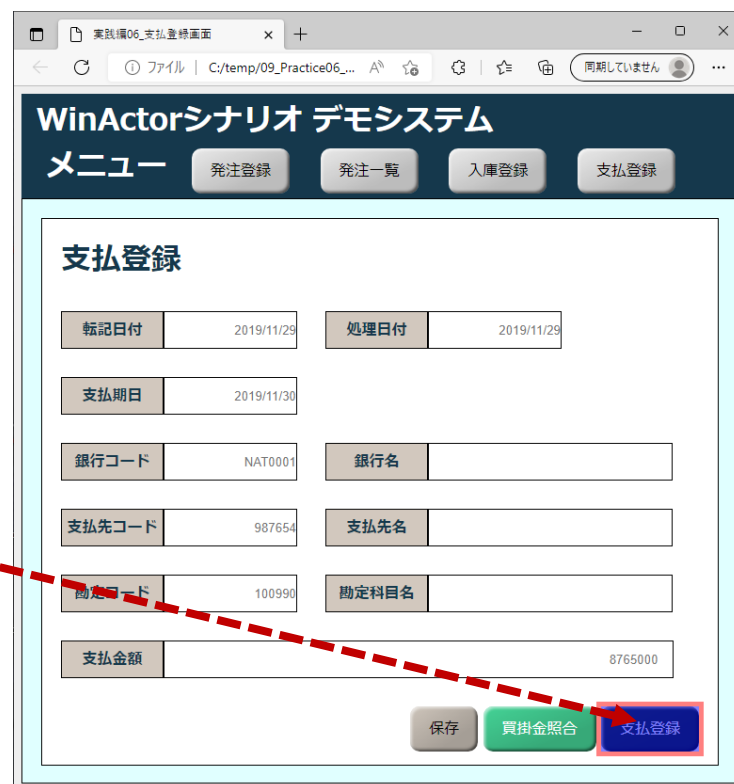
6-7

登録設定

15 <要素のPath> の<選択> ボタンをクリックします。



16 <支払登録> ボタンにマウスカーソルを移動し、色が変わったら、クリックします。



6

条件によって処理内容を変更する

6-7

登録設定

17 <更新> ボタンをクリックします。

プロパティ

Python実行

名前 クリック

コメント

設定 スクリプト 注釈 バージョン情報

未設定の場合、10,000ミリ秒となります。

ブラウザ名 値⇒

ウィンドウ識別名 実践編06_支払登録画面-プロフ: 

Path種別 XPath

要素のPath 値⇒ //*[ ブラウザ 選択 候補

選択するフレーム トップフレーム

フレームのPath 値⇒

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値⇒

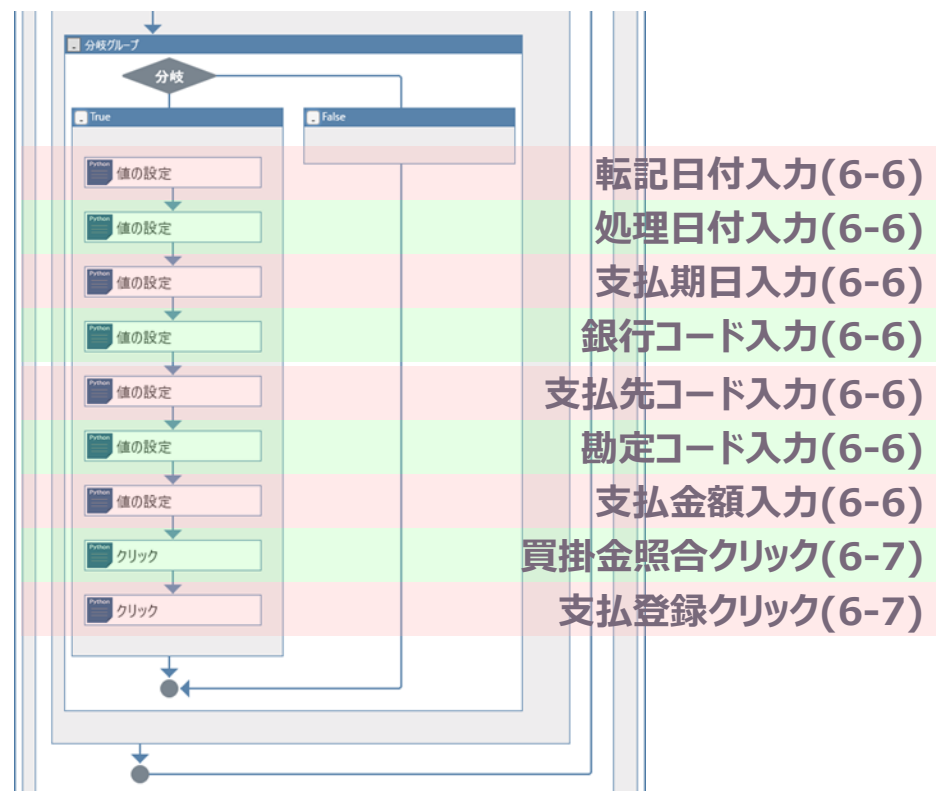
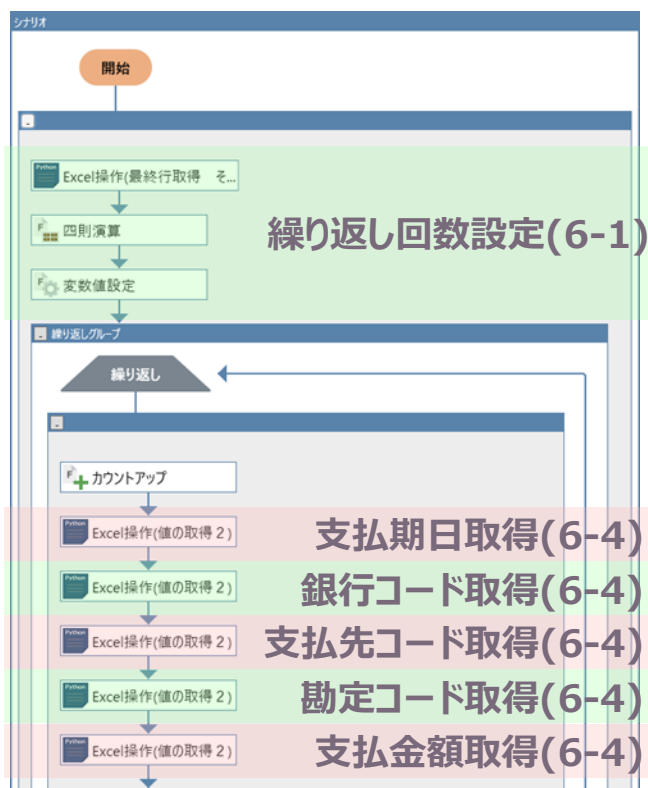
更新 元に戻す

6

条件によって処理内容を変更する

6-8 シナリオ全体の確認

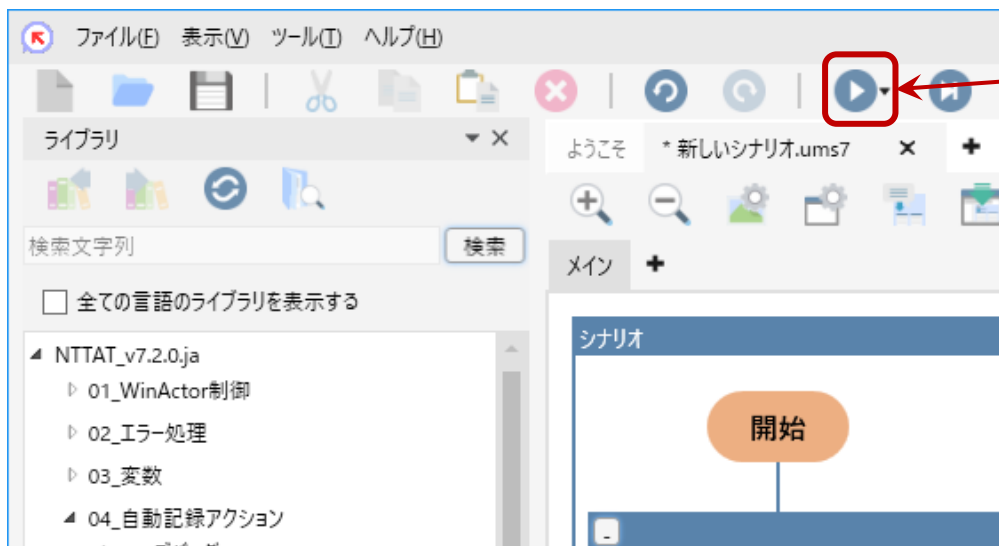
1 作成したシナリオが以下の通りになっていることを確認します。



6

条件によって処理内容を変更する

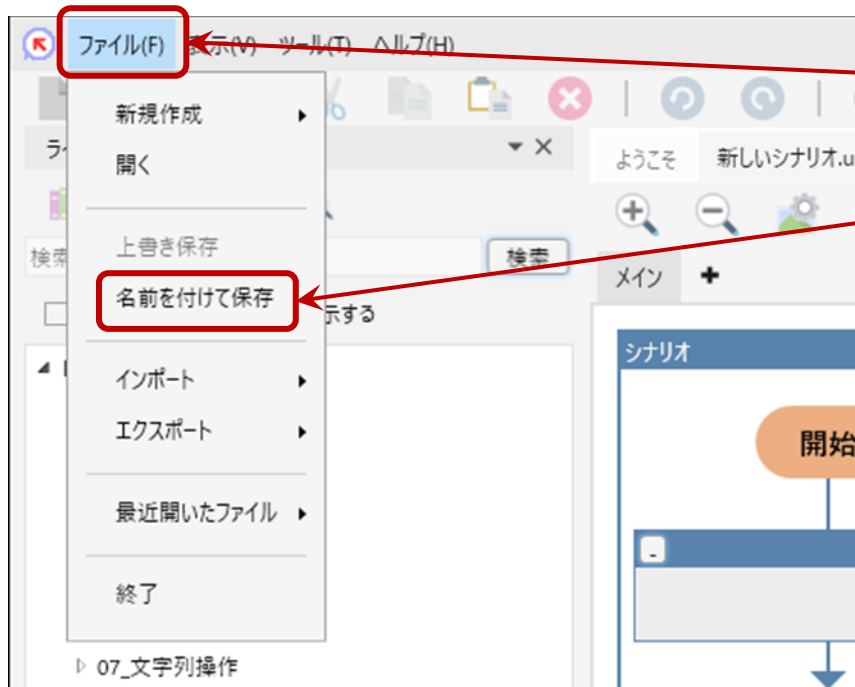
6-9 シナリオの実行



1 <シナリオ実行> ボタンをクリックし、エラーなくシナリオが進むか確認します。

2 エラーが発生したら、エラーメッセージを確認しプロパティ等の設定を見直します。

6-10 シナリオの保存



1 <ファイル> メニューをクリックします。

2 <名前を付けて保存> メニューをクリックします。

3 保存先を選択し、ファイル名を付けて保存します。



すでに保存されているファイルについては、<上書き保存> から保存することができます。



「宇院さん、基礎編と実践編を学んでみてどうでしたか？」



「実践編のおかげで、実際の業務に使えるシナリオが作れそうです！ 繰り返し処理は動きが理解しやすかったですし、分岐処理についてはとても設定しやすく分かりやすかったです！」



「それは良かったです！ 基礎編含めこれまで学んだ内容は、WinActorの基本になりますので是非おさえてください！ これから実際の業務で活用しながら、他の機能にもチャレンジしていきましょう！」

「シナリオ作成の参考になるページのURLを記載しますので、ご活用ください。」

内容	タイトル	URL
WinActor公式サイト	WinActor® 業務効率を劇的にカイゼンできる純国産RPAツール	https://winactor.biz/
フリーのサンプルシナリオの提供	サンプルシナリオ	https://winactor.biz/sample-scenario/
シナリオに組み込めるライブラリの提供	プチライブラリー一覧	https://winactor.biz/library/
認定研修	RPAツール「WinActor®」の認定研修	https://winactor.biz/winactor-certification.html
Q&A	WinActor よくあるご質問	http://www.matchcontact.net/winactor_jp/

Point!



～ 応用編 ～



1

シナリオ作成 ～応用編プロローグ～

導入

最近、宇院さん以外にもWinActorを使用して業務の自動化を進めているようです。そのため、他の人が作ったシナリオの修正や、トラブル対応をする必要が出てきました。今回はシナリオ作成から少し進んだ、修正しやすいシナリオ作成の方法やトラブル時の対応方法について学んでいきましょう。



宇院(ういん)さん

調達部で働く2年目社員。
WinActorをだいぶ使いこなせるようになり、他の人に教えることができるようになった。



阿久田(あくた)さん

調達部で働く1年目社員。
宇院さんを見て自分もWinActorを使い始めた。

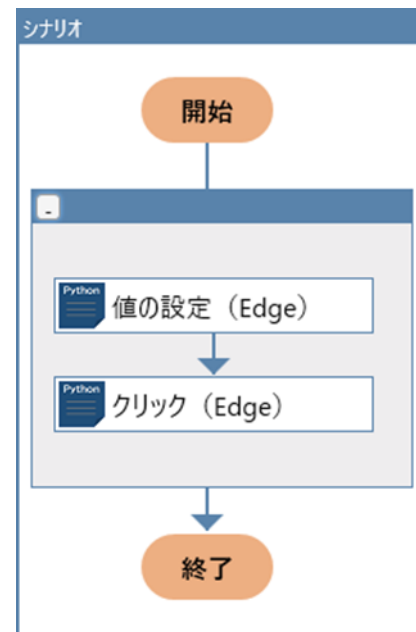


増田(ますた)先輩

既にWinActorを使用している他部署の先輩社員。
基礎編、実践編に続き宇院さんのサポート役をしている。



自動でシナリオ作成



自動記録機能によるシナリオ作成について学ぶ

導入

阿久田さんは最近WinActorを使い始めましたが、シナリオ作成に手間取っているようです。そこに宇院さんが様子を見にやってきました。



「阿久田さん、シナリオ作れそうですか？」



「うーん、中々難しいですね…やりたいことは分かるんですがノードやライブラリがたくさんあってどれを使えばいいのか分かりません。」



「それならシナリオを自動で作ってくれる自動記録という機能を使ってみませんか？私も使ったことが無いので一緒に覚えましょう！」



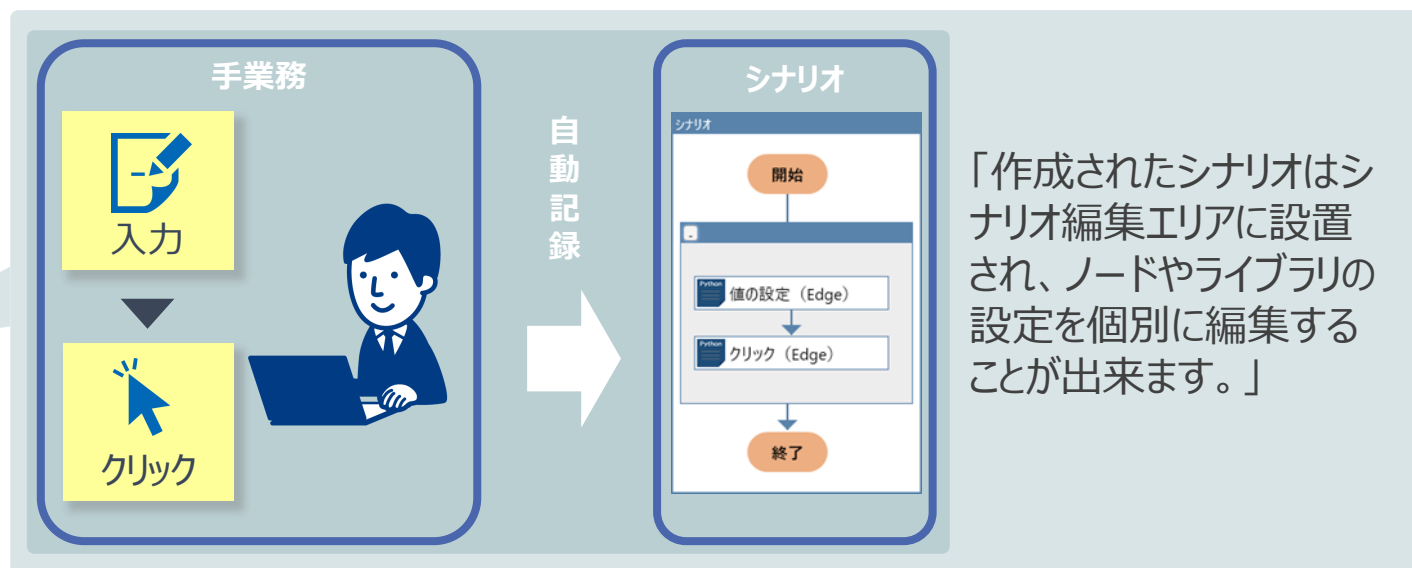
「宇院さんは初めからノードを使用してシナリオを作っていましたが、今回は、人が行った操作を自動でシナリオ作成できる自動記録という機能を使ってみましょう。」



「＜自動記録＞は具体的にどのような機能ですか？」

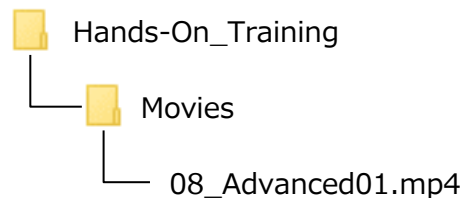


「自動記録とは、**利用者の操作を自動で記録し、シナリオを作成する機能**です。いつも行っている業務と同じようにマウスやキーボードを操作するだけで、シナリオが作成されます。」





「それでは実際に自動記録の使い方を見てみましょう。
以下フォルダのファイル名をダブルクリックして再生してください。」



「自動記録はシナリオ作成を効率化出来る機能ですが、繰り返しや分岐処理ができないため、自動記録で作成されたシナリオは全体のベースとして利用したり、シナリオの一部として使用するなど**シナリオ作成の補助として使用するのがよい**でしょう。」

Tips

自動記録時のエラー

**自動記録が出来ない場合**

いくつかの条件下で自動記録を行った際に、シナリオが作成されない（空のグループが作成される）ケースがあります。

以下にケースと対応方法を記載しますが、利用者のみでは解決が難しいケースが多い為、情報システム部などパソコンを管理している部署の方に相談をお願いします。

ケース①：記録しようとしているアプリケーションが管理者の権限で起動している場合

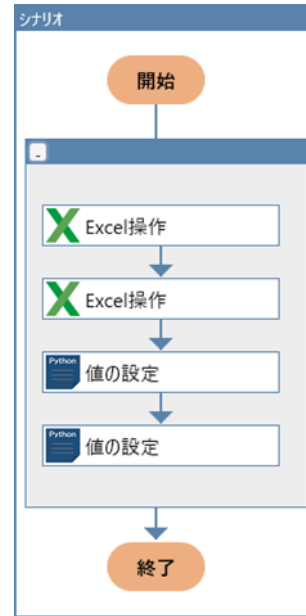
対応方法：可能ならばアプリケーションを一般権限で起動するか、WinActorを管理者権限で起動してください。

ケース②：記録しようとしているアプリケーションが保護モードになっている場合

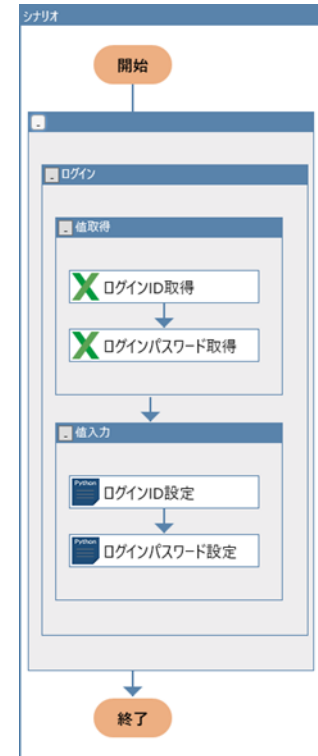
対応方法：EdgeのIEモードやAcrobatReaderなどが当てはまります。設定から保護モードを無効にしてください。

ケース③：記録しようとしているアプリケーションがリモートデスクトップ上で起動している場合

対応方法：自動記録ではシナリオを作ることが出来ません、エミュレーションノードと画像マッチングを組み合わせて利用ください。



修正



修正しやすいシナリオ作成方法を学ぶ

導入

阿久田さんが作ったシナリオがなぜか動きません。
見かねた宇院さんが助け舟をだしました。



「ちょっとシナリオ見せてごらんよ…。なんかごちゃごちゃしてて見づらいね…。」



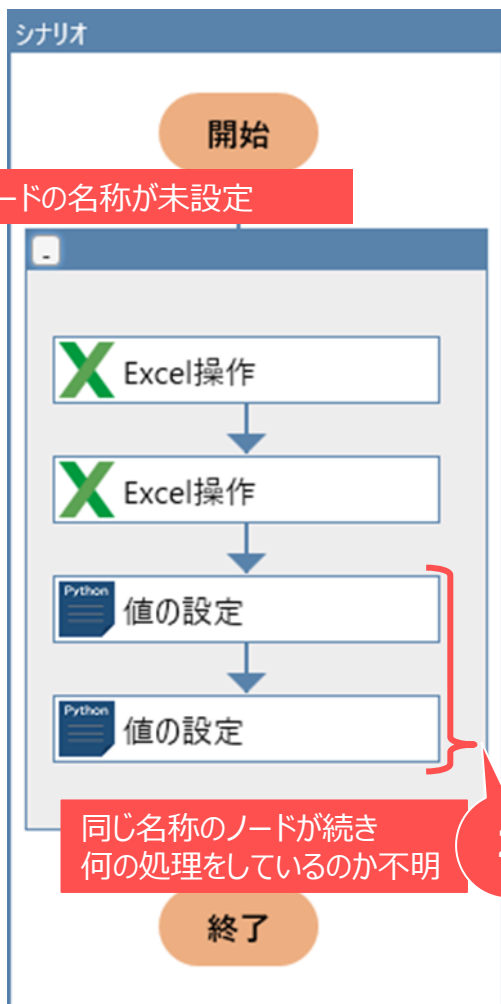
「私は分かるんですけど…見づらいですか？」



「もうちょっと分かりやすいシナリオじゃないとサポートできないな…。これは私にも言えることだけど、シナリオを作る時に気を付けないといけないことだよ。」



「シナリオを作成するためにはノードやライブラリを使いこなすことも大事ですが、他の人が見やすいように作成することや、後から修正がしやすいように作成することなど気を付けるポイントはたくさんあります。今回はそういったポイントを学びましょう。」



「さすがにこれは見づらいんじゃないの？ 一つ一つプロパティを開いていかないと何の動作をしているのか分からないよ…」

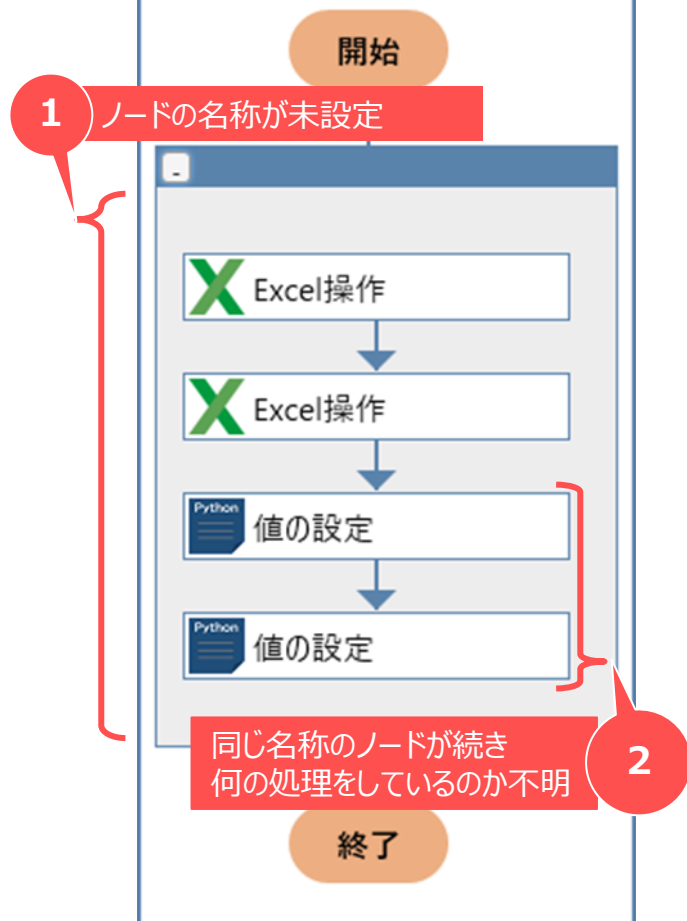


「そういわれるとそんな気がしてきました…」

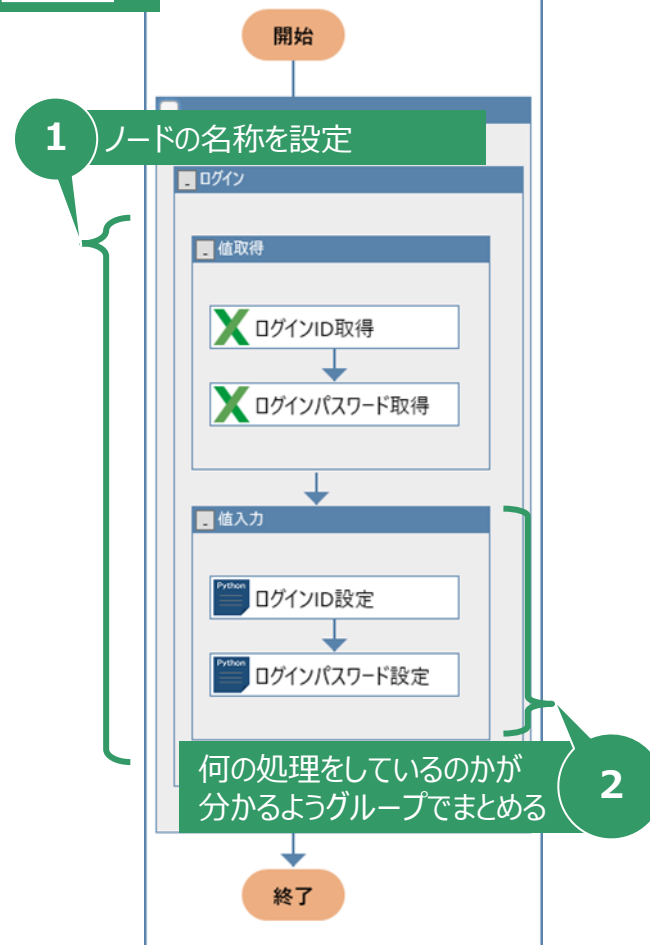


「①どのノードがどんな動きをしているのか分かりづらいですね、②同じ名称のノードが続いているのも見づらい原因になっています。作成した本人では気づきにくいものですが、もう少し見やすく出来そうですね。次のような形に修正してみましょう！」

Before



After





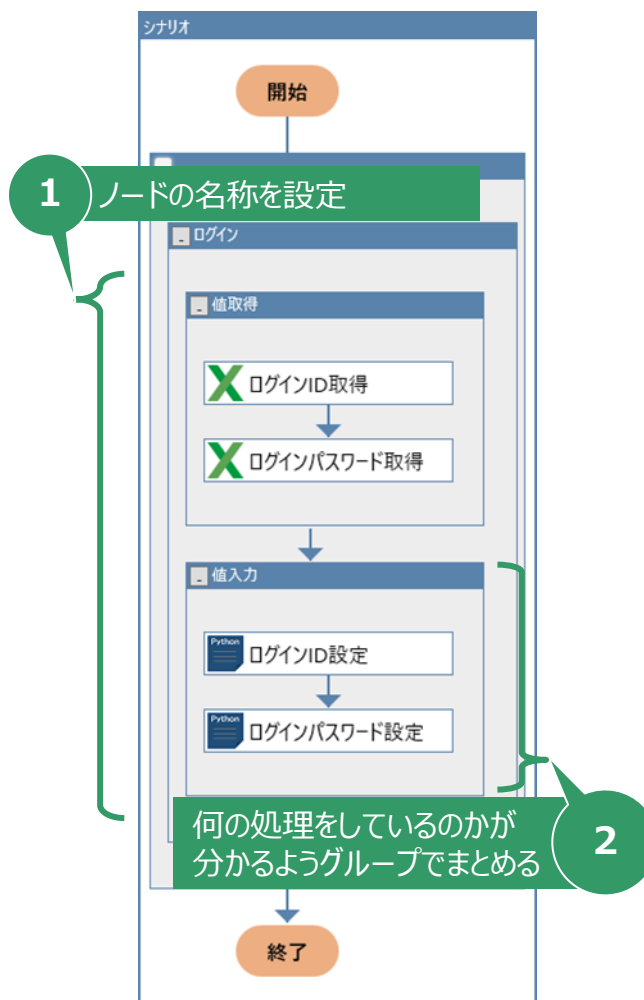
「あ！見やすいです！」



「さきほどのシナリオと比べて明らかに見やすいですね、これならシナリオを作っていない私にも分かりやすいです！」



「非常に分かりやすくなりましたね！これは
①ノードの名称を変更することでノードの動きを分かりやすくし、処理毎に②ノードをグループでまとめることでどの処理の動きなのかを見やすくしています。次のスライドからこのようなシナリオの作り方を説明しますので、阿久田さんだけでなく宇院さんも一緒に学んでいきましょう！」





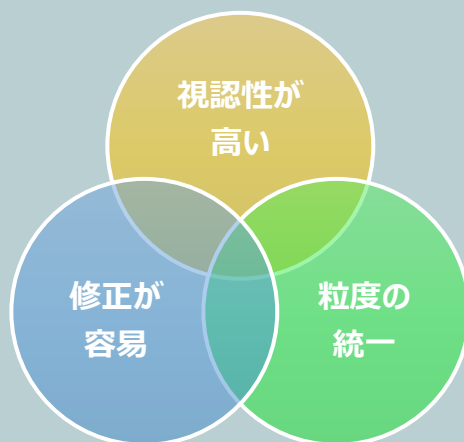
「分かりやすいシナリオのポイントって何があるんですか？」



「いくつかありますが、例として「視認性が高い」「修正が容易」「粒度（記載の細かさ/粗さ）の統一」などがあります。これを総じて保守性が高いと言います。」



分かりやすさのポイント例



「保守性が高いと以下のようなメリットがあります。」

- 他者に説明しやすくなることでシナリオの共有が可能になる。
- トラブル時に迅速な修正が可能になる。
- 新しい業務を自動化する際に、既存シナリオのノードやライブラリの流用が可能になる。



～Column～

保守性とは？

保守性とは、以下のようなことを指します。

- 異常の発見・修正のしやすさ
- 機能の追加・修正のしやすさ
- 異常なく動いている物の維持のしやすさ

元々は「保守」という、正常な状態を維持できるように手入れする意味から来ており、WinActorの場合は、シナリオの誤りや不具合の発見・修正のしやすさ、機能追加などの行いやすさ、シナリオの見やすさなどが中心となります。メンテナンスのしやすさとも言い換えられます。

シナリオを動かし始めると様々な課題が発生しますが、保守性を高くすることによって、効率的にストレスなく、解決することが可能です。



「保守性が高いシナリオを作るにあたって具体的には何をすればいいんですか？」



「まずはシナリオ作成する際の**ルールを決めることが大事**です。ルールを決めることで以下のような良い流れを作ることが出来ます。」

1. ルール（共通認識）を決めることで
2. シナリオや書類に記載する言葉の粒度が統一され
3. 保守性が高くなる

「今回は基本的な以下のルールを説明していきます。」



1. シナリオの保存先とシナリオ名の命名規則を決める
2. ノード/ライブラリの名称は動作が分かる名称に変更する
3. ノード/ライブラリはグルーピングする
4. ノードのプロパティに直接文字や値を入力せず、変数と外部ファイルを利用する

「説明の最後にルールテンプレートを配布しますので、テンプレートを元に自分たちでルールを作ってみましょう！」

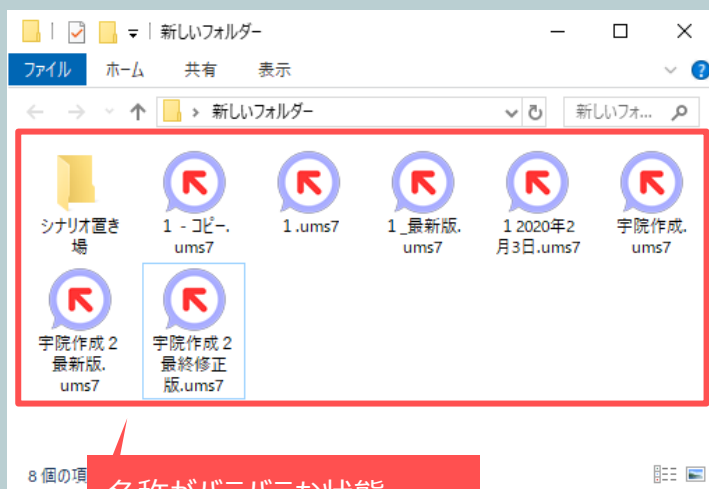


「“ 1. シナリオの保存先とシナリオ名の命名規則を決める” は、保守性を高めるための基本中の基本です。」



Point!

「シナリオに限らず、複数人で業務を進める場合にファイル名やフォルダ名、ファイル保存場所がバラバラだと効率が悪いですね？ WinActorにおいても**シナリオの保存場所や名称はルールを決めた方が効率良くシナリオを作ることができます。**」



「効率が良いだけでなく、**誤って他のシナリオを修正してしまったり、同じ動作のシナリオを作ってしまうなどのミスも防ぎやすくなります。**」



「“ 2. ノード/ライブラリの名称は動作が分かる名称に変更する” については、以下のような手順で行います。」

Point!



「対象のノードをダブルクリックし、表示されたプロパティの＜名前＞を動作を表した名前に変更します。変更するとシナリオ編集エリアのノードの表示が変更されます。」

プロパティ

Excel操作

名前 ログインID読み取り

コメント

操作 値の取得

取得元

ファイル名 変数名もしくは値を選択

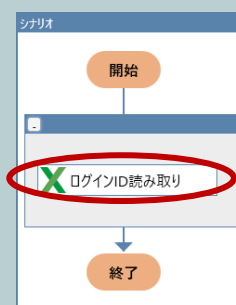
シート名 変数名もしくは値を選択

セル位置 変数名もしくは値を選択

格納先

変数 変数名を選択

更新 元に戻す



「＜Excel操作＞ など同じノードを複数設置することがあると思うのですが、その場合などは名前を変更することによりシナリオ編集エリアでの視認性がグッと上がります。」

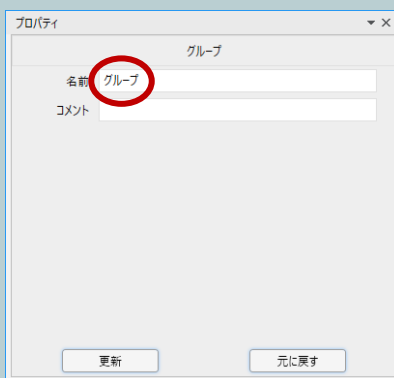
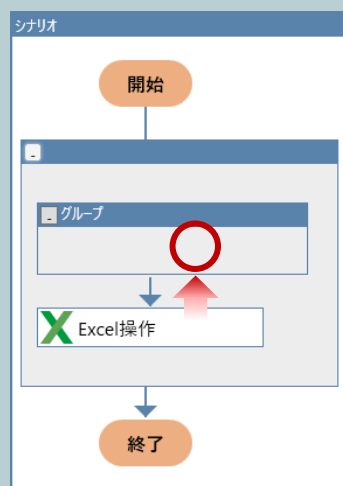


「“3. ノード/ライブラリはグルーピングする”については、以下のような手順で行います。」

Point!



「グループノードをシナリオにドラッグした後に、グルーピングしたいノードをグループノード内にドラッグします。こうすることでログイン処理など処理単位でグルーピングすることができ、視認性が高くなります。」



「ノードやライブラリと同様に、グループもプロパティの<名前>を変更することで、シナリオ編集エリアでのグループ名の表示が変わります。」

Tips

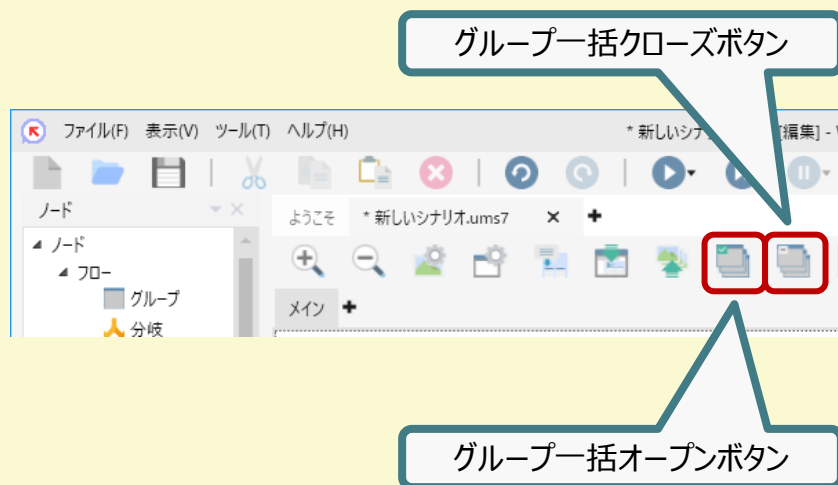
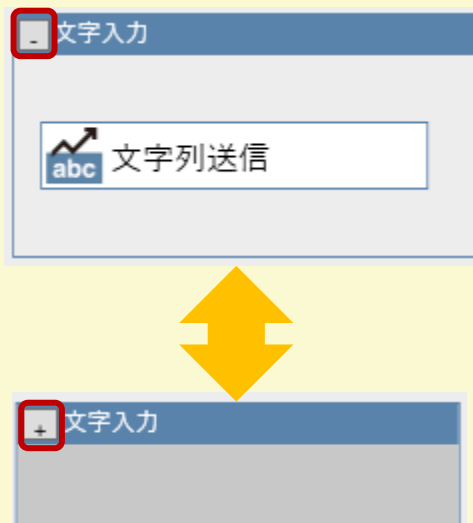
グループの閉じる、開く



グループを閉じることで複数のノードやライブラリをコンパクトに表示出来ます。

ノードやライブラリをまとめたグループは、左上のボタンで閉じて小さくしたり、開いて元の大きさに戻すことが出来ます。

シナリオが長くなりグループの数が増えた場合は、グループ一括オープンボタンとグループ一括クローズボタンを使って、見やすい状態にしてシナリオを作っていきましょう。





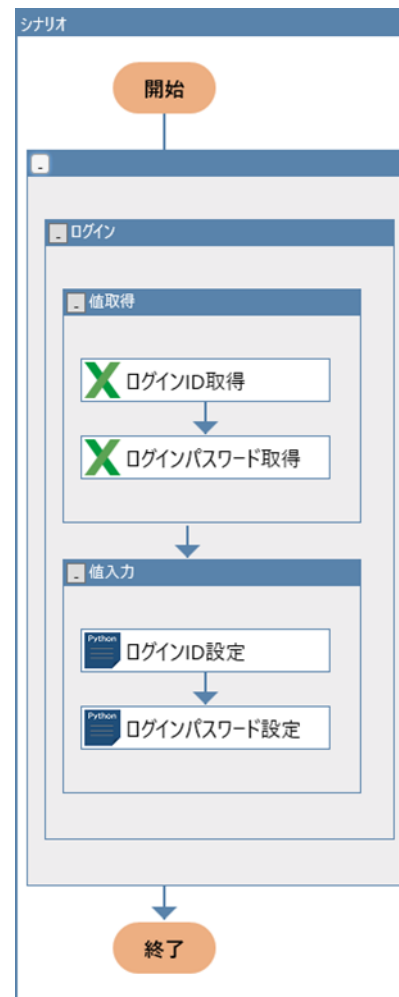
「ここまでのルールをさきほどのシナリオに適用してみると見やすいシナリオになりましたね！」



「一目で動きが分かったと、わざわざプロパティを開いて設定を確認する手間が省けますね。」



「ノードの名称変更とグルーピングでシナリオの見やすさは改善されましたね！次は修正のしやすさについて改善しましょう！」





「“ 4. ノードのプロパティに直接文字や値を入力せず、変数と外部ファイルを利用する” について詳細を説明しますね。」

Point!



「これまで作成したシナリオでは、ノードやライブラリのプロパティに直接ログインIDや入力ファイルの場所を設定していましたが、この方法だとログインIDやファイルの場所が変わってしまった場合に、シナリオを開いて一つ一つ手で修正する必要があります。これでは保守性が高いとは言えませんよね。」

取得元

ファイル名	値⇒ C:\temp¥04_Basics05_Hacchu_	...
シート名	値⇒ Sheet1	...
セル位置	変数名もしくは値を選択	

値が直接入力された状態

「値が変更されたとしても、シナリオ自体の修正は必要なく、設定ファイルを修正するだけで済むようにしましょう！」



「具体的には以下のような設定ファイルを作成しましょう。例としてログインIDを入力する処理を行います。まず、値が修正される可能性がある情報を設定ファイルとしてExcelファイルに記載しましょう。ここで重要なことは実際にシナリオで使用する値と、その値を何に使っているかを記載することです。」

Point!



「このようにシナリオとは別のファイルとして作成することで、**シナリオを修正する際に誤って他の設定を修正してしまうミスを防ぐ**ことが出来ます。」

設定ファイル				
通番	設定内容	詳細説明	シナリオで読み取る値	シナリオでの変数
1	ログインID	システムにログインするためのID	User01	ログインID
2	ログインパスワード	システムにログインするためのパスワード	Password	ログインパスワード
3	発注管理台帳の場所	登録時に使用する発注管理台帳の保存場所	C:\temp¥04_Basics05_Hacchu_Kanri_Daichou.xlsx	発注管理台帳場所

設定ファイルにシナリオで読み取る値を記載

「そして、前述のとおり値を修正する際もこの設定ファイルを修正するだけでよいのです。」

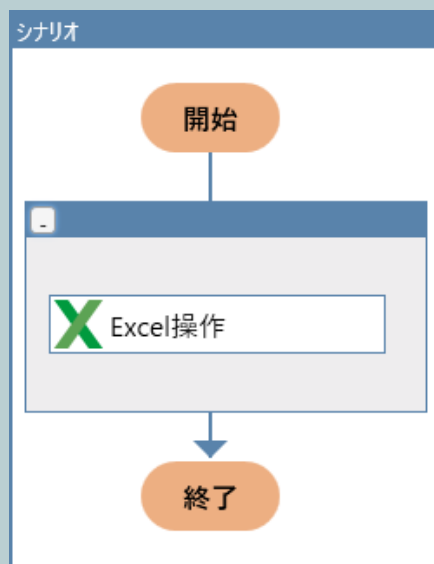


「次に作成した設定ファイルをWinActorに読み込ませてみましょう！＜Excel操作＞ノードを使用し、設定ファイルに記載されているログインIDを読み込みます。」

Point!



「今までのExcelを読み込む手順と同じように、プロパティを設定していきます。」



ここに設定ファイルの保存場所と、読み込む値が記載されているシート名とセル位置を記載

変数「ログインID」に値を読み込む

3

シナリオ作成 ～応用編. 2～

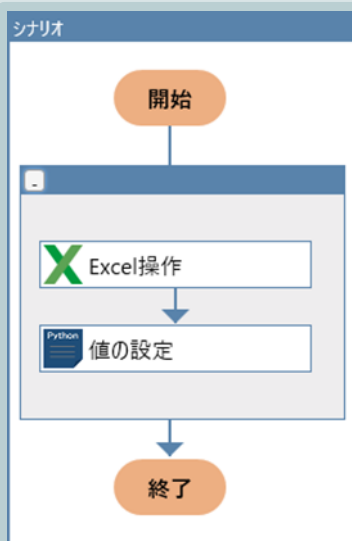


「次に読み込んだ値を使用しましょう！ <値の設定> ライブラリを使用し、さきほど読み込んだ変数を設定します。」

Point!



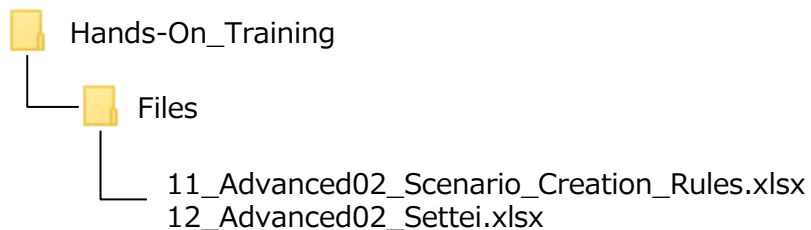
「この手法を使えば、ログインIDが変わった場合でも、シナリオを修正することなく、設定ファイルだけを修正すればよい保守性の高いシナリオになります。」



値を読み込んだ変数「ログインID」を設定

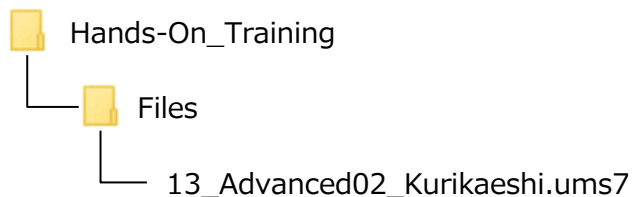


「それでは私が用意したテンプレートをベースにルールを作成してみましょう！以下に保存されているルールテンプレートをベースに、オレンジ色の箇所を考えてみてください。テンプレートそのまま使用しても大丈夫です。」





「ルールは出来ましたか？次に作成したルールを“実践編1 何度も同じ処理を繰り返す”にて作成したシナリオに適用してみましょう！また、参考までに下記フォルダにルール適用済ファイルを用意してあります。」





「どうですか？ぱっと見ただけでもルール適用前と適用後でだいぶ保守性に差があると思いませんか？」



「そうですね、ルールを決めて揃えるだけでこんなに見やすくなるなんて思いませんでした！」



「これで私も阿久田さんのシナリオを直せそうです！」



「保守性は、複数人でシナリオを作るとなると必須になる技術です。今回のルールを決めるというのも一つの手段であって、それ以外にも保守性を高める方法があります。ポイントをおさえて保守性の高いシナリオを作っていきましょう！」



シナリオが動かない



データが入力されていない



ファイルが出力されていない

トラブル時の対応方法を学ぶ

導入

阿久田さんが作ったシナリオが突然動かなくなっていました。阿久田さんの支援で宇院さんがサポートに入りますが…



「今まで動いていたシナリオが突然動かなくなっていました！」



「とりあえずシナリオを開いて見てみよう…シナリオの途中で止まっているね、原因は何なんだろう…？」



「業務で使用するファイルが出力されていないと、他部署から連絡がきてます…どうしましょう…」



「トラブル時の対応は誰でも慌てるものです。今回はトラブルが起きた時の対応と、同じトラブルを起こさないための対策について学びましょう」



「トラブルが起きた時には何をすればいいのでしょうか？」

「大きく分けて以下の3つの対応が必要です。」

1. シナリオが動かないことで起きている事象を確認する（事象の確認）
2. なぜシナリオが正常に動かないのか原因を見つける（原因の調査）
3. 正常に動くようにシナリオの修正などの対策を行う（対策の実施）

トラブルが起きた時の対応の流れ



Point!

「この3つを順序立てて行くと、トラブル対応をスムーズに進めることができます。次のスライドで詳しく見ていきましょう。」



「“1. 事象の確認”については、まず何が起きているかを確認します。」

「事象の例としては＜出力されるべきファイルが出力されていない＞、＜社内システムに入力されるべき情報が入力されていない＞、＜必要な書類が印刷されていない＞、＜メールが送信されていない＞などが考えられます。」

起きる事象の例



ファイルが出力されて
いない



情報が入力され
ていない



印刷がされてい
ない



メールが送信され
ていない



転記がされてい
ない



処理結果がおか
しい

「実際に発生する事象は1つだけではなく、複数発生していることが多いです。事象の確認ができなければ原因の調査を正しく進めることが出来ません。焦らず事象を把握しましょう。」

Point!

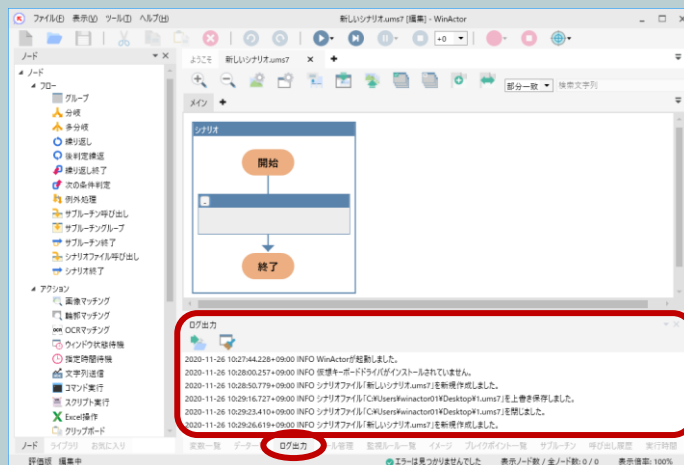




「次に“ 2. 原因の調査”として**シナリオ実行ログを確認**します。」

「ログとはWinActorが動作する際に記録される動作情報であり、ノードの実行日時や実行された内容などの事柄が記載されています。」

Point!



「今回のような**トラブル時**にも**ログ**は記録されており、正常に動作しなかった**ノードID**や、動作しない**理由**などが記載されています。」



「ログから原因となっているノードが判明した後は、対象のノードのプロパティを確認しましょう。」

「プロパティに記載されているファイルパスやマッチング画像は正しいですか？いつの間にか変わってしまっていたりしませんか？シナリオ自体に修正を行っていないのに正常に動作しなくなった場合は、まずは以下のような**シナリオ以外の原因**を疑いましょう。」

シナリオ動作に影響を与える外部要因の例



「シナリオ以外の原因というのは、

- ・ シナリオを実行するパソコンが変わった
- ・ 入力ファイルの保管場所（フォルダ）が変わった
- ・ 画像マッチングで使用している社内システムの画像が変わった

など**シナリオ自体の問題ではなく外部環境の変化によるもの**です。」

Point!





「原因が判明したところで、“3. 対策の実施”としてシナリオが正常に動くように直していきます。対策には暫定対策と恒久対策がありますので、まずは暫定対策として、**シナリオを動くように修正**しましょう。暫定対策とは応急手当のような意味でシナリオを動かす為の＜取り急ぎ＞の対応です。」



「具体的には何を対応すればよいのですか？」



「トラブルは千差万別のため、全ての対応方法をここで説明することは難しいのですが、基本的にはログを確認し、内容を読み解き、対応を行いましょう。次のスライドで良く発生するトラブルとその対応をまとめましたので、こちらもご確認ください。」

Point!



「以下は一例になります。詳細はWinActor操作マニュアルを参照ください」

エラーログ	原因	対応
ファイルの読み込みに失敗しました。不正なファイルパスです。	Excel操作ノードのプロパティで指定した読み込みファイルが無い。	設定した場所にファイルがあるか確認し、無い場合はファイルがある場所を設定しなおす。
ウィンドウ識別名 * * * * に一致するウィンドウが存在しませんでした。	ノード/ライブラリであらかじめ設定したアプリケーションが起動していない。	アプリケーションを起動し再実行する。
スクリプトの実行に失敗しました。 エラー番号:0x000000001 内容：指定されたファイルを開く事ができません。	パスワードがかけられたExcelファイルを、Excel操作などExcel関連ノード/ライブラリを使用して開こうとした。	Excelファイルのパスワードを解除する。
スクリプトの実行に失敗しました。 エラー番号:0x000000001 内容：指定されたシートを開く事ができません。	指定したExcelファイルのシート名が誤っている。	Excelファイルのシート名を正しい値に設定する。
スクリプトの実行に失敗しました。 エラー番号:0x000000001 内容：この名前は既に使用されています。別の名前を入力してください。	Excelファイルのコピー元シート名とコピー先シート名の値が同一であるため。	コピー先シート名の値を変更する。
スクリプトの実行に失敗しました。 エラー番号:0x000003ec 内容：WorkbookクラスのSaveAsメソッドが失敗しました。	Excelファイルの保存先を正しく指定できなかった。	Excelファイルの保存先に同名のファイルが無いか確認し、存在した場合は削除する。
スクリプトの実行に失敗しました。 エラー番号:0x0000000d 内容：型が一致しません。	ノード/ライブラリへ入力した値に誤りがある。 (例：日付として成立しない年月日など)	入力した値を修正する。



～Column～

シナリオの保守性とトラブル対応の関係

シナリオの保守性が高ければ高いほどトラブル対応はしやすくなります。

シナリオの見やすさ分かりやすさは、トラブルが起こった際に行う以下 3 つの対応のしやすさと直結するからです。

1. 事象の確認
2. 原因の調査
3. 対策の実施

特に対応の速さが必要になる暫定対策を行う際には非常に重要な要素になります。

出来る限りトラブルが起こりにくいシナリオを作成することが一番なのですが、起こってしまった場合にすぐ対応できるシナリオを作ることが大事です。



「次に恒久対策として**同じトラブルが起きない様に修正**をしましょう！再発防止とも言います。ノードやライブラリのプロパティ修正の他に＜監視ルール＞の適用や＜例外処理＞の追加などが有効な場合もあります。」



「＜例外処理＞？＜監視ルール＞？どちらも今まで使用したことがないですね…」



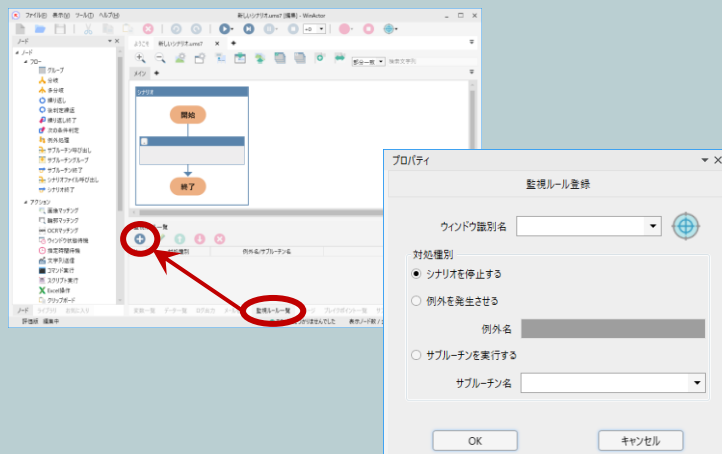
「どちらもシナリオを正常に動かすための強力な機能です。次のスライドで＜監視ルール＞と＜例外処理＞の機能紹介をしますので、ご確認ください。」



「監視ルールとは、シナリオ実行中に**事前に設定したウィンドウ名の画面が表示されると、実行中のシナリオの停止や別の処理を実施させることが出来る機能**です。」

「監視ルールを設定することで、他のアプリケーションが画面に表示された時やダイアログが表示された時に、シナリオを正常に止めることや別の処理でシナリオを続けさせることが出来ます。」

Point!



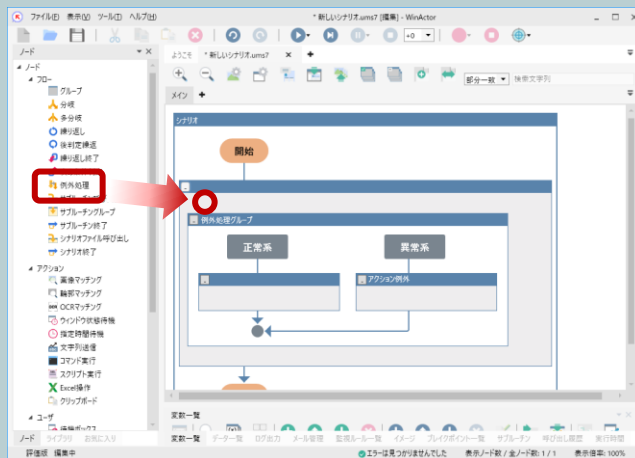
「WinActor以外に、Outlookやチャットアプリなど常時動いているアプリケーションがあるパソコンや、外部要因でダイアログが表示される可能性がある環境では設定しておいた方が良いでしょう。」



「監視ルール他に予期せぬエラーが出た場合の処理として、＜例外処理＞ノードがあります。例外処理はシナリオ全体に影響する監視ルールと違い、＜例外処理＞ノードの中に設定したノード/ライブラリが対象になります。」

「正常系と異常系というの2つの処理を持っており、通常時に実行される**正常系**の中に入っているノード/ライブラリの処理中にエラーが発生した場合、正常系の処理は中止され**異常系**の処理が動くようになっています。」

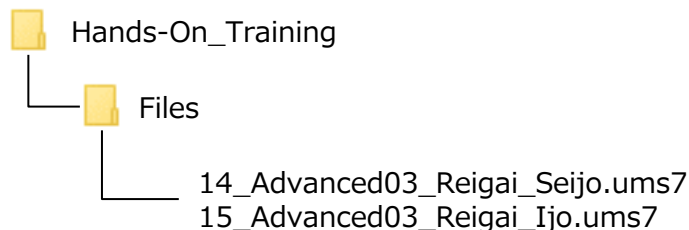
Point!



「通常時の処理は正常系に入れておき、エラーが発生した時の処理を異常系に入れておく使い方が良いでしょう。」



「＜例外処理＞については、サンプルのシナリオを動かしてみて動作を確認しましょう。」

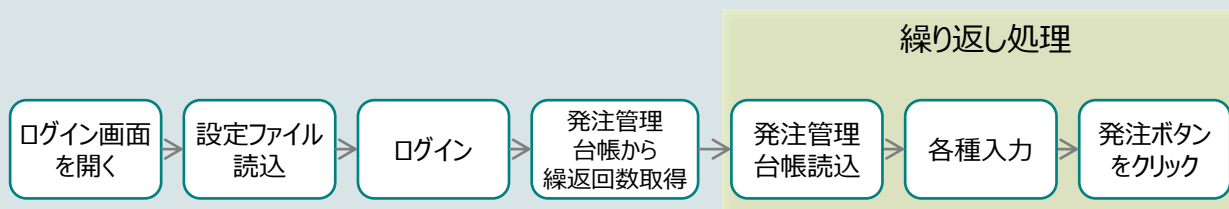


「正常時シナリオと異常時シナリオの動作を見比べると、異常時動作のエラー発生時には、異常系の処理に切り替わることが分かります。このサンプルシナリオの正常系に、今まで作成したシナリオをコピーして動作させてみるのも良いですね。」



「これまで学んだ内容を踏まえて、トラブルが発生して動かないシナリオを修正してみましょう！まずは正常に動いていたシナリオの説明をしますね。」

「正常に動いていた際は、以下の流れで動いていました。」

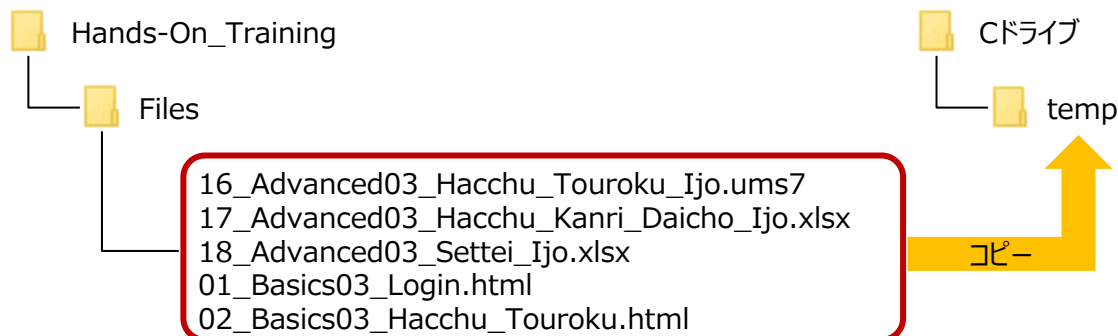


「使用しているファイルは以下になります。」

種類	ファイル名
シナリオファイル	16_Advanced03_Hacchu_Touroku_Ijo.ums7
設定ファイル	18_Advanced03_Settei_Ijo.xlsx
発注管理台帳	17_Advanced03_Hacchu_Kanri_Daicho_Ijo.xlsx



「それでは、以下に保存されている“動かないシナリオや設定ファイル” を<C:¥temp>に保存してからシナリオを実行してみましょう。」



「いくつかエラーが発生し正常に動かないかと思います。今まで学んだ内容を思い出してシナリオを直していきましょう！」



「どうですか？ 困った場合は下記をチェックしてみるといいでしょう。」

エラーログ	実際のトラブル	確認チェックポイント
スクリプトの実行に失敗しました。 エラー番号: 0x00000001 内容: 指定されたシートが見つかりません。	「発注管理台帳に入力されている列数読み取り」で「指定されたシートが見つかりません。」とエラーが表示される	発注管理台帳ファイルのシート名は正しいですか？ 台帳利用者がシート名を変えてしまったのかもしれない。確認してみましょう！
スクリプトの実行に失敗しました。 エラー番号: 0x00000001 内容: ライブラリ実行（spv_Web）に失敗しました。 ウィンドウ識別ルール「基礎編 03_発注登録画面-プロファイル1-MicrosoftEdge」にて指定されるブラウザが見つかりません。	ログインIDが入力されない	表示されている画面は発注画面まで遷移していますか？まずはログインが出来るようにしましょう！ ログインIDは設定ファイルに記載があります。 設定ファイルを見直してみましょう！
	ログインできない	画像マッチングの設定は正しいですか？ プロパティのマウス操作位置の設定を確認してみましょう！

Point!

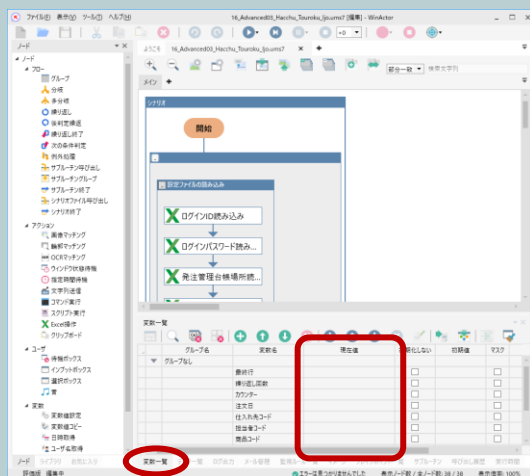




「トラブル対応の際に、現在の変数の中に入っている値を確認したいことがあるかと思います。そういった場合は、変数一覧画面の＜現在値＞を確認してみましょう。」

「＜現在値＞は、シナリオの動作と連動しており、リアルタイムで更新されています。Excel操作など**変数の中身が変わるノードを実行した後に、変数がどう変わったのか確認する際に便利な機能**です。」

Point!



「自動で入力される値が、思っていた値と違う場合などは、フローチャート画面やメイン画面を見ても原因が分かりません。シナリオがうまく動かない場合は変数一覧画面も確認してみましょう。」



「シナリオの修正は終わりましたか？正常に動くシナリオは以下に保存されていますので自分が修正したシナリオと見比べてみましょう。」



「どうでしたか？うまく修正が出来なかった箇所については、正常に動くシナリオを参考にしながら自分で手を動かして修正してみましょう！」



「トラブル対応お疲れ様でした。いつものシナリオ作成とは違う技術が必要でしたがどうですか？」



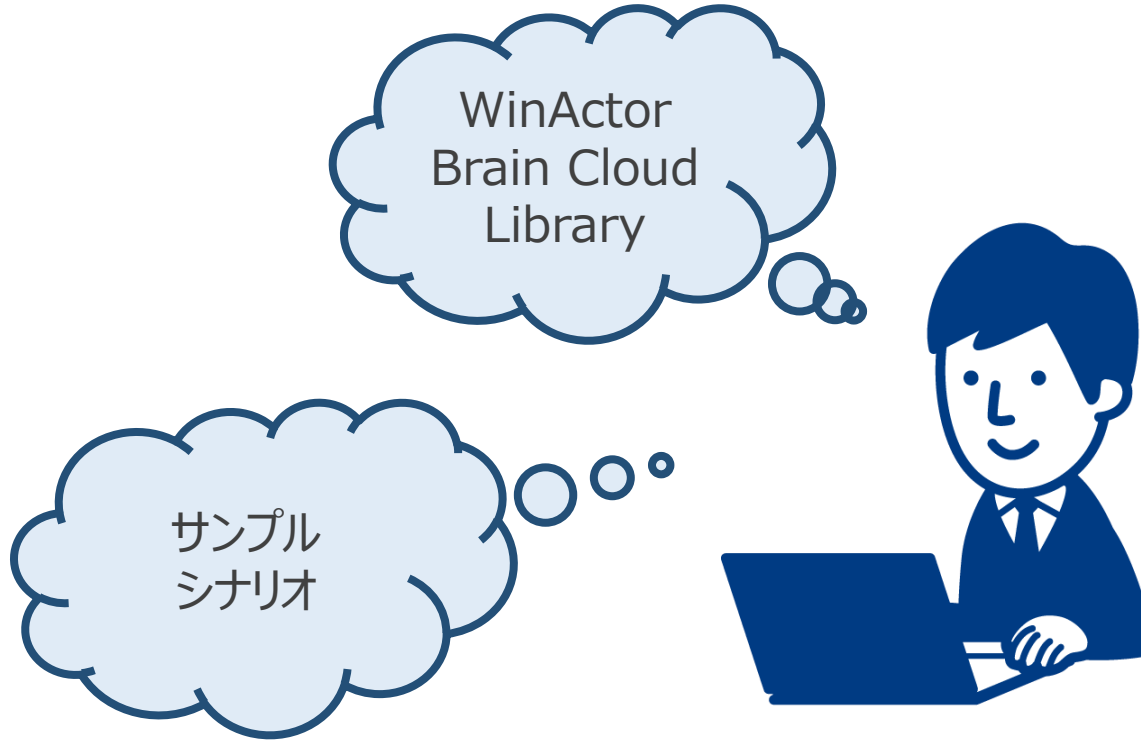
「難しいですね…慌ててしまって頭がこんがらがってしまいます…」



「私はコツが分かればなんでしょう！よく考えてみるとトラブル対応を行う際にも前回学習した保守性を高める工夫が効いてくる訳ですね！」



「その通りです、さすが宇院さんですね！トラブル時の対応は、迅速さを求められることが多々あります。今回のように座学で知識を学んだ後、実際に手を動かしてシナリオを修正することでより現場で活かせる知識になります。慌てず臆せず対応していきましょう！」



WinActorの知識を広げる

導入

今まで増田先輩に頼ってきた宇院さんですが、もっとWinActorについて学びたいと思い始めました。どういった手段があるのでしょうか。



「今まで私は増田先輩に頼りきりだったけれども、今は私が先輩の立場になっている。もっと自分自身でWinActorを学べるようにしたいんですが…」



「宇院さんもすっかり先輩ですね。WinActorにはいろいろなコンテンツや、より便利に使用するためのツールがありますので紹介します。」



「インストーラーに同梱されている資料以外にもあるんですか？」

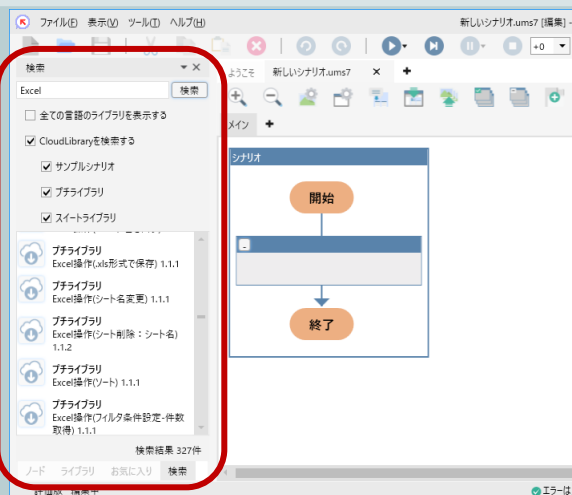


「もちろんです！ WinActorにはまだいろいろな機能があります。すべてを使いこなす必要はないですが、どれも業務に利用できる機能ばかりです。次のスライドで案内しますね。」



「まずは、WinActor Brain Cloud Libraryです。」

「WinActor Brain Cloud LibraryはWinActorのシナリオ作成をより簡単に、より効率的に行うための、クラウド上で動作するシナリオ作成支援ツールです。提供機能の一つとして、WinActor同梱ライブラリとWinActor公式サイトで公開しているプチライブラリを検索し、ダウンロードすることができます。」



「プチライブラリには、指定した日付が平日か休日か判定するライブラリやExcelファイルを印刷するライブラリなど実業務で使えるライブラリが多数登録されています。」



「プチライブラリはシナリオを作るためのライブラリですが、サンプルシナリオという形でシナリオも公開しています。」

「WinActor公式サイトにはサンプルシナリオ一覧というページで、すぐに使用できるシナリオファイルを公開しています。例として複数のExcelファイルからデータを抽出して集計するサンプルシナリオや、ウェブページをスクロールしながら画像マッチングを行うサンプルシナリオなどがあり、こちらも上手に使っていきましょう。」



「WinActor Brain Cloud Libraryやサンプルシナリオの詳細については、以下のURLにアクセスしてください。」

WinActor公式サイト
(<https://winactor.biz/>)



「宇院さんと阿久田さん、応用編を学んでみてどうでしたか？」



「シナリオ作成が出来るようになって少し天狗になっていましたが…シナリオ作成以外にも保守性を高めることやトラブル対応など学ぶことがありますね。」



「私はシナリオ作成を深く学ぶ前に、応用編の内容を学ぶことが出来て得した気分です！」



「それは良かったです。今回学んだことはWinActorを複数人で使用していく場合に非常に重要な内容になっています。保守性を高めることは最初手間に感じるかもしれませんが、作成したシナリオが自分の手を離れ他の人が修正することを考えると、少しでも分かりやすくしておきたいですね。この思いやりがトラブル時の迅速な対応に繋がります。これからはこういった観点も頭においてシナリオを作っていきますよ！」



宇院さん、阿久田さん、増田先輩たちと基礎編、実践編、応用編を通じて色々なことを学んできましたが、WinActorにはもっと色々な機能や使いやすいツールがあります。

前述したWinActor Brain Cloud Libraryや公式ホームページに掲載されているサンプルシナリオなどを使い、WinActorで業務をより効率化していきましょう！

機能について追求することも大事ですが、WinActorの合言葉は「現場フレンドリーな業務改善を実行するパートナー」です。今回の宇院さんや阿久田さんのようにまずは現場で手を動かしてみることがなによりも大事になります。まずは「この業務WinActorで出来るかも？」という小さい気づきから始めて、シナリオを作ってみましょう！その思いをWinActorは全力でフォローします。

おつかれさまでした！

更新履歷



版	作成・変更日	変更内容
1.0	2020/01/18	初版作成
1.1	2020/03/25	応用編の追加
1.2	2020/08/20	記載内容の修正
1.3	2020/08/27	記載内容の修正
1.4	2020/12/25	WinActor7.2への対応、記載内容の修正
1.5	2021/03/03	Tipsの追加、記載内容の修正
1.6	2021/10/29	Microsoft EdgeのIEモードへの対応、記載内容の修正
2.0	2023/09/08	Microsoft EdgeとGoogle Chromeへの対応
2.1	2025/05/01	WinActor7.6への対応、記載内容の修正
2.2	2025/03/26	WinActor7.7への対応、記載内容の修正



本書は著作権法上の保護を受けています。本書の一部あるいは全部を無断で複写、複製することは禁じられています。
本書の内容は予告なく変更される場合があります。

WA7-T-20260609