

RPAテキスト

2024年版



RPAの基礎知識を習得し、
実業務を自動化できるスキルを得る

- RPAの概要
- Power Automate for Desktopの概要と基本機能
- Excel編
- ブラウザ編
- UIアプリケーション編

RPAの概要



Robotic Process Automation

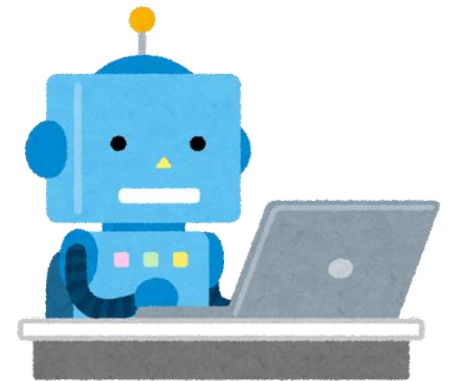
(ロボティック・プロセス・オートメーション) の略

人間がコンピューター上で行っている定型作業をロボットで自動化すること。
画面の内容を確認・複写して入力する作業などを人の代わりに自動実行します。

ミスなく24時間365日稼働してくれて残業代も発生しません。

プログラミング言語を利用しない、

ノーコード・ローコード開発手法が特徴です。



① RPAに適した業務

電子データ
での作業

誰がやっても同じ
結果になる作業

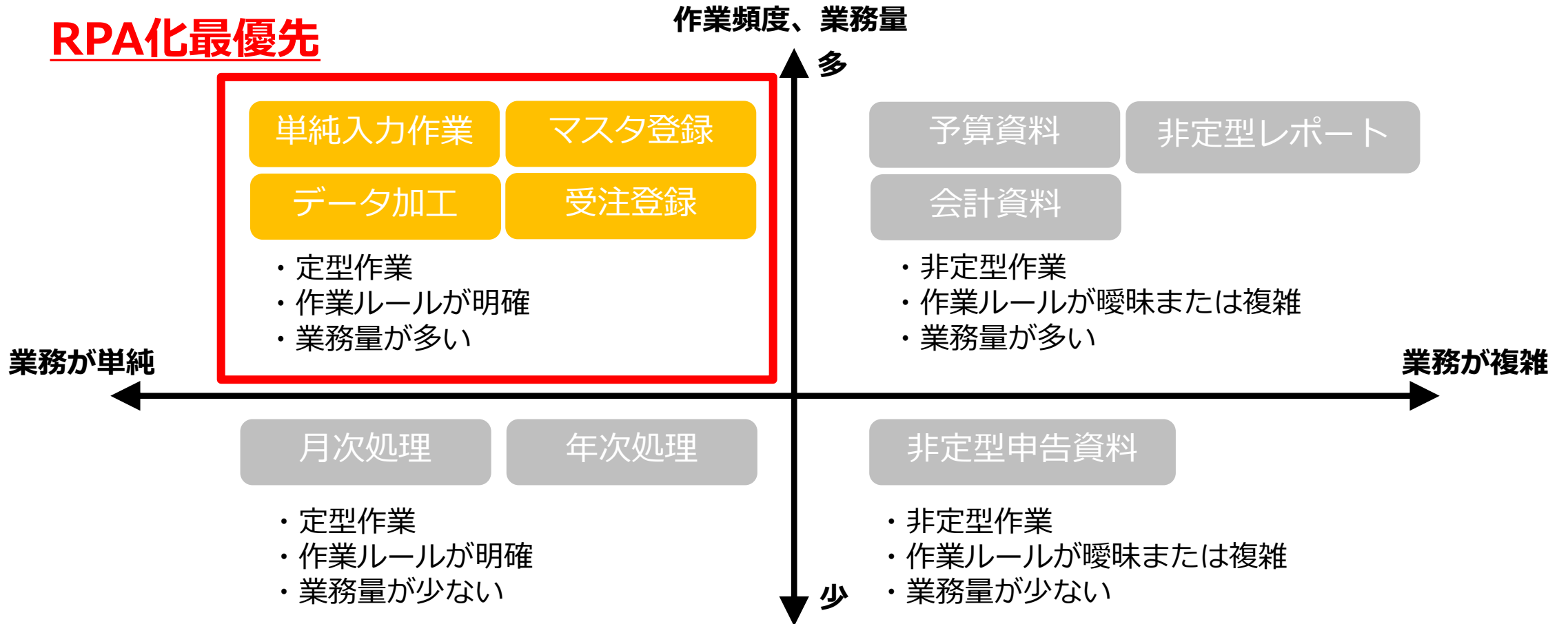
頻度・業務量
が多い作業

① RPAに適した業務

業務名	作業内容
請求書処理	請求書のスキャン、データの抽出、システムへの入力
顧客対応	個別の顧客ニーズに合わせた対話や交渉
顧客データの更新	定期的な顧客情報の更新と整理
戦略立案	多角的な分析と判断を要する戦略の策定
マーケティングコンテンツの作成	オリジナルコンテンツの企画・制作
売上報告の自動作成	売上データから自動で報告書を生成

① RPAに適した業務

RPA化最優先



① RPAの種類

大きく分けて3種類あります。

デスクトップ型

低コスト
運用開始が容易

・

全体管理が難しい

サーバー型

高性能
全体管理が可能

・

高コスト

クラウド型

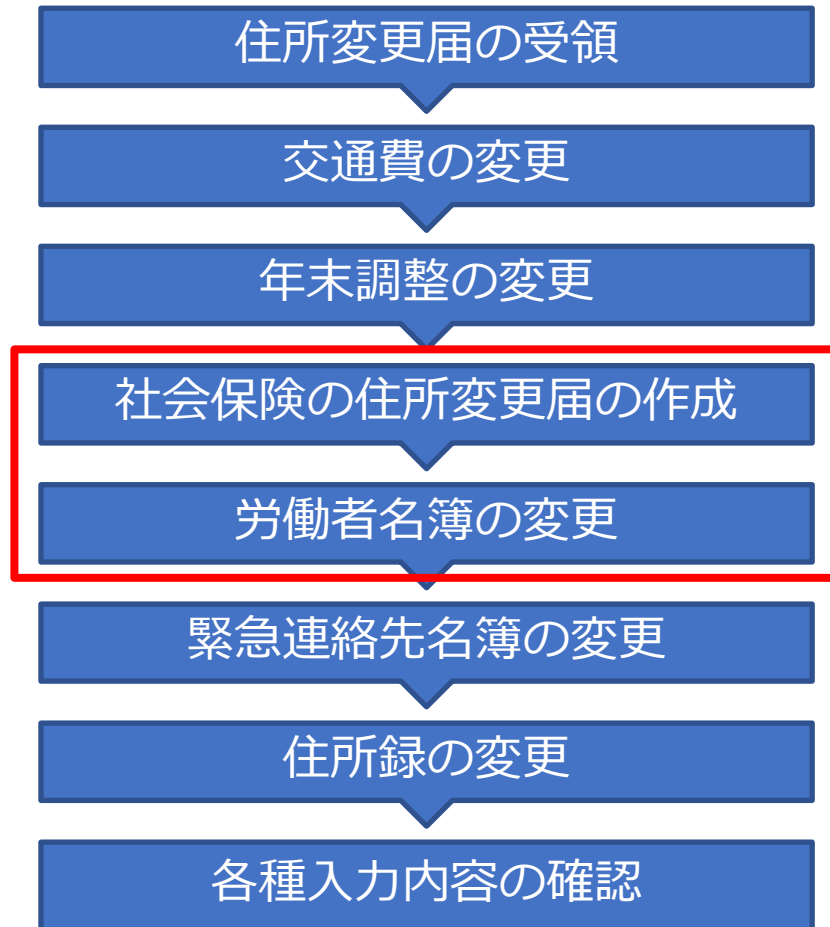
インストール不要

・

デスクトップの
自動化ができない

① RPA導入で心折れないためのポイント

すべての工程をいきなりRPA化せず、業務を細分化して自動化し、少しずつロボットを育てていくことが成功のカギになります。



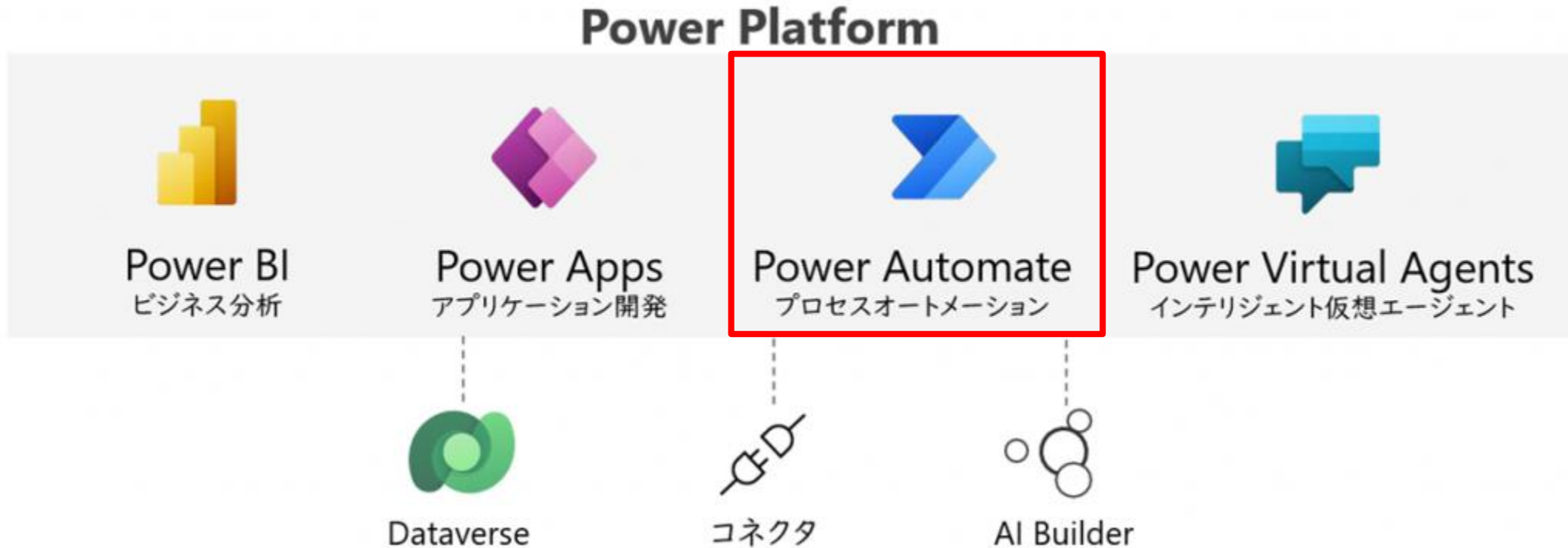
まずはここだけ

Power Automate for Desktopの概要と基本機能



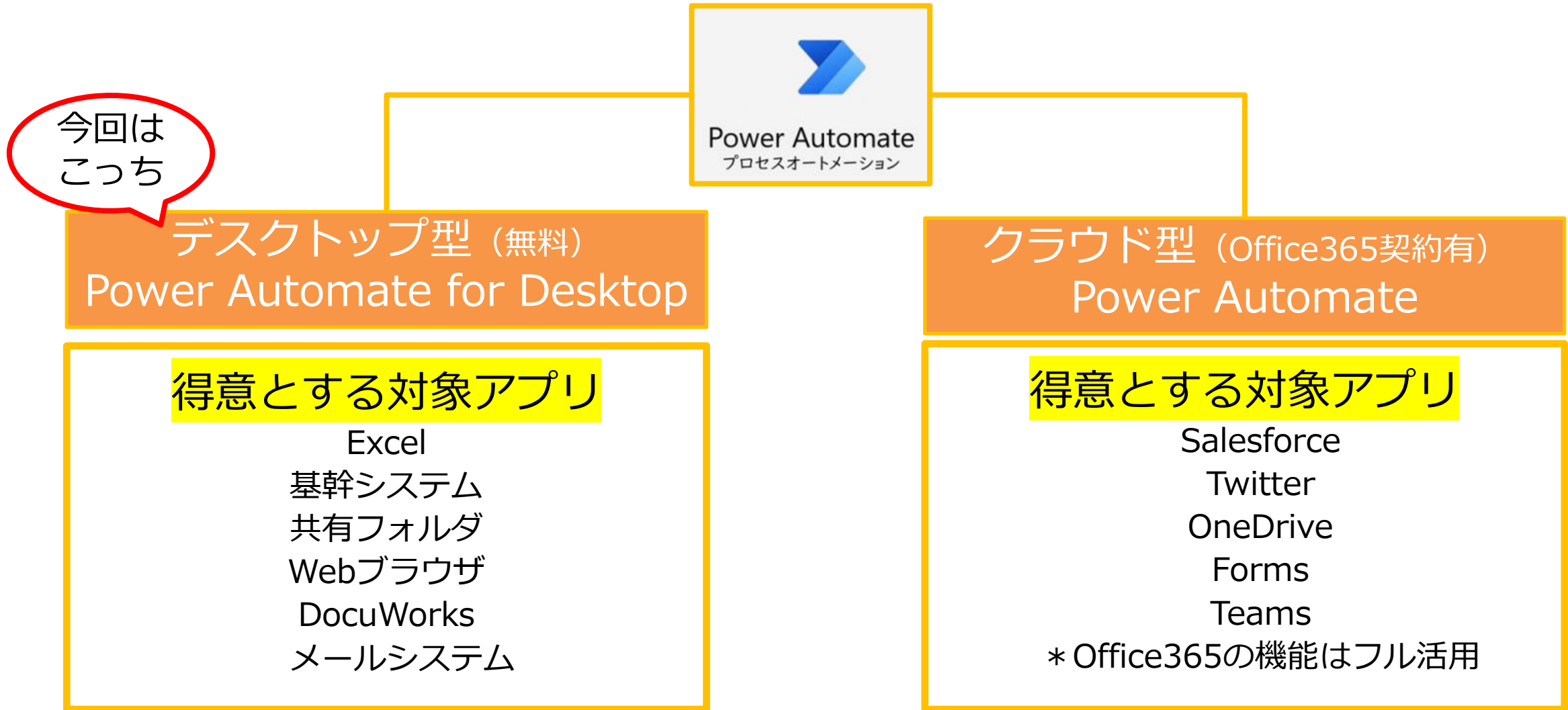
① Power Automate for Desktopとは

Microsoftが提供するRPAです。ローコード（少ないプログラムコード）開発できるツールのため、エンジニアでなくても業務を自動化できます。



① Power Automate for Desktopとは

Power Automateは2つの機能に分けられます。



① 画面構成

フローの新規作成

The screenshot shows the Power Automate web interface. At the top, there is a blue header with the text 'Power Automate' and a user profile '松尾 匠'. Below the header, there is a navigation bar with a '+ 新しいフロー' button and search options. The main content area is titled '自分のフロー' and contains a table of flows. The table has columns for '名前' (Name), '変更日時' (Modified Date), and 'ステータス' (Status). Two flows are listed: 'test01' and 'test02'. The 'test02' flow is selected, and its action menu is highlighted with a red box. The action menu contains icons for '再生' (Play), '停止' (Stop), '編集' (Edit), and a vertical ellipsis (More options).

名前	変更日時	ステータス
test01	5 分前	現在編集中
test02	5 分前	現在編集中

作成したフローのリスト

メニュー「再生、停止、編集」

① フロー作成画面

グループごとにすべてのアクションが機能ごとにグループ分けされています

アクションをこのスペースに配置しフローを作成していきます

アクション

アクションの検索

- > 変数
- > 条件
- > ループ
- > フローコントロール
- > フローを実行する
- > システム
- > ワークステーション
- > スクリプト
- > ファイル
- > フォルダー
- > 圧縮
- > UIオートメーション
- > HTTP
- > ブラウザー自動化
- > Excel
- > データベース
- > メール
- > Exchange
- > Outlook

サブフロー Main

ここにはまだアクションがありません

開始するには、このペインにアクションをドラッグ アンド ドロップするか、レコーダーを使用してアクションを取得します

変数

変数の検索

入出力変数 0

ここにはまだ入力または出力の変数がありません

フロー変数 0

表示する変数がありません

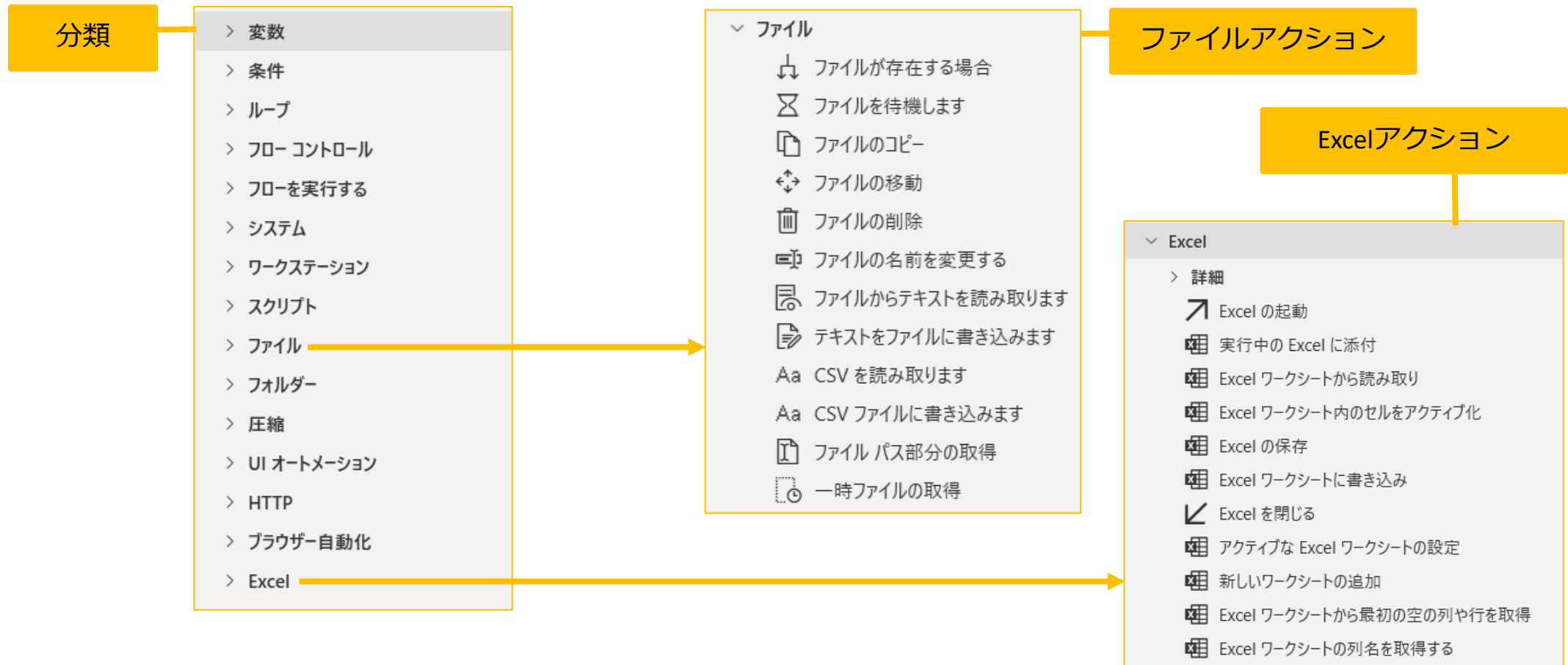
ステータス: 準備完了

0 選択されたアクション 0 アクション 1 サブフロー 実行遅延 100 ミリ秒

変数に格納された値の確認や変数名の変更ができます

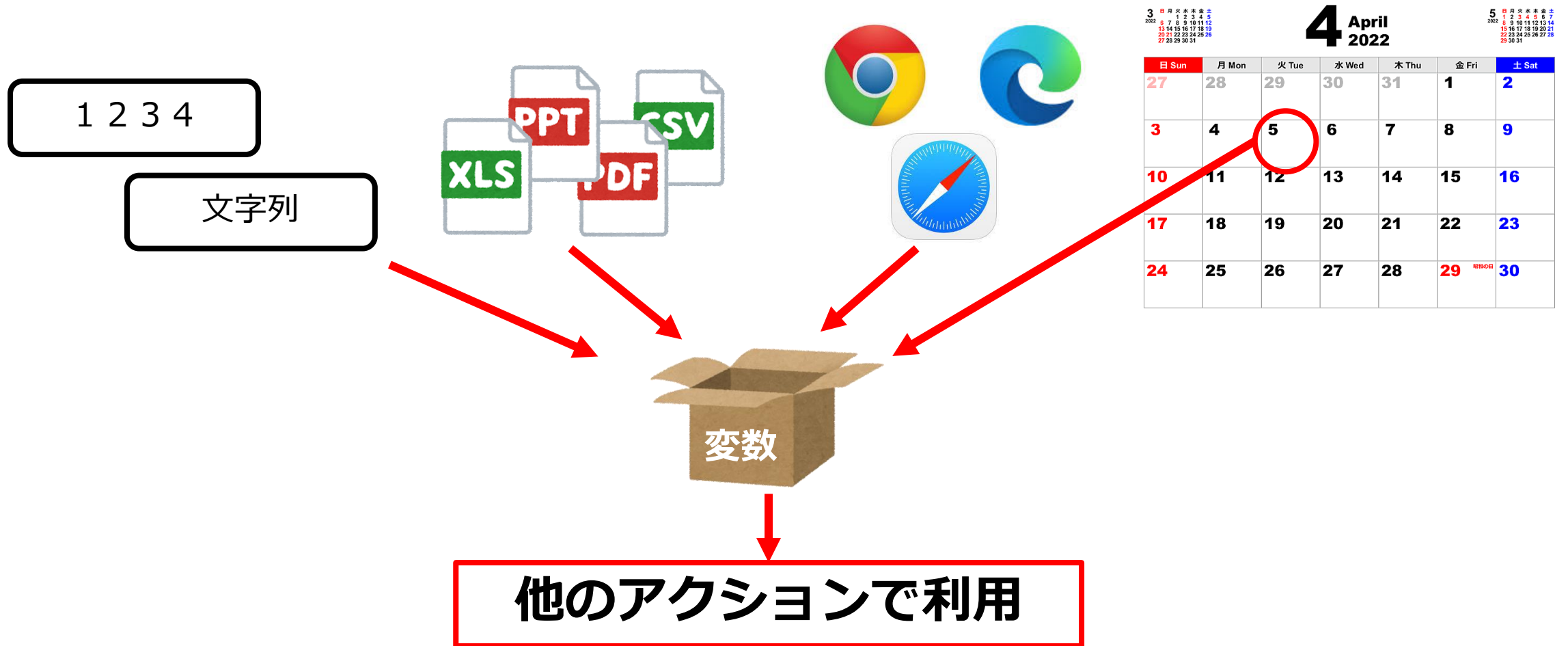
① アクション

「RPAにさせたい操作」の部品がアクションと呼び、これをつなぎあわせてフローを作成します。アクションは操作対象ごとに分類されています。すべてを覚える必要はなく、自動化したい業務から徐々に覚えていきましょう。



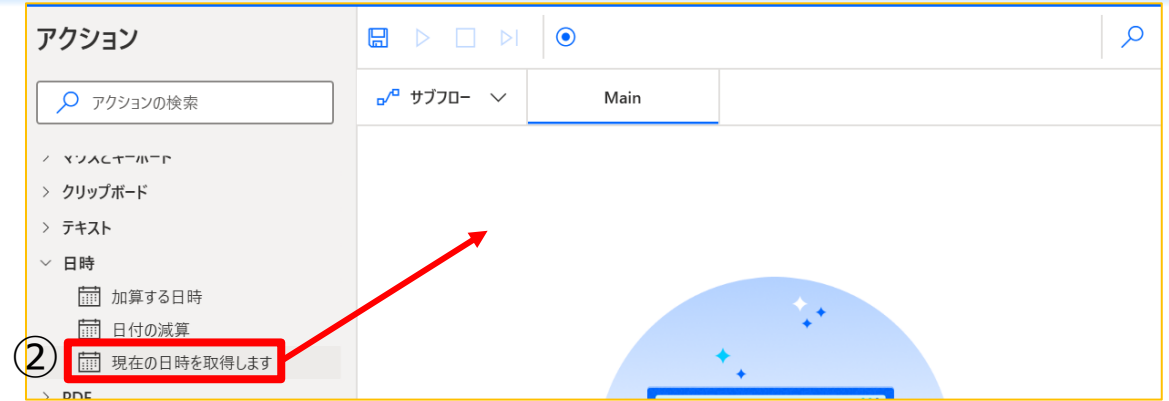
① 変数

フローの中で利用される値は、**変数**として設定されます。
他のアクションで何度も利用するときのためにとても重要なものです。



① [演習 1]変数を理解する

- ① +新しいフローより新規作成画面を開きます
- ②日時→現在の日時を取得します
をワークスペースにドラッグorダブルクリック



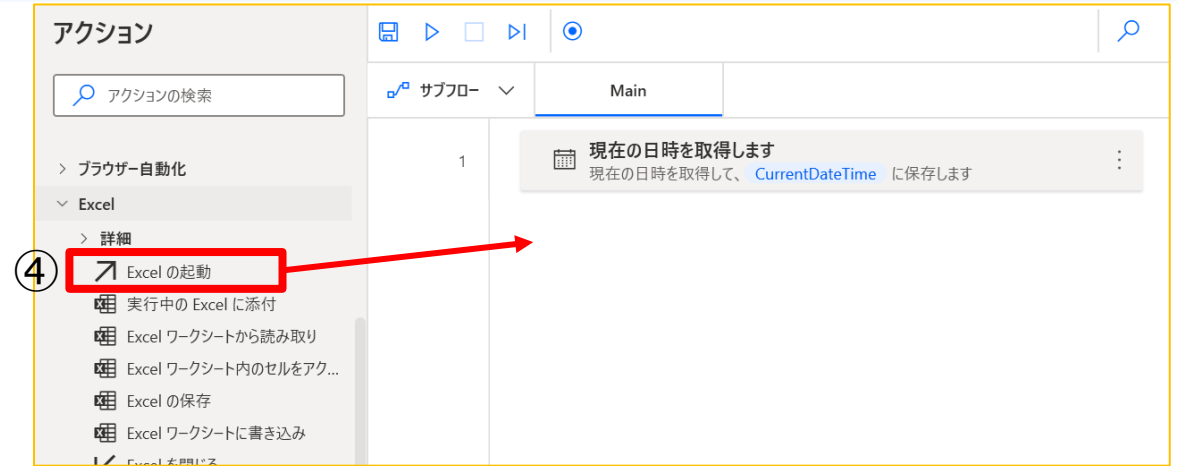
- ③保存

変数が自動で生成されます



① [演習 1]変数を理解する

④ Excel→Excelの起動
をドラッグorダブルクリック



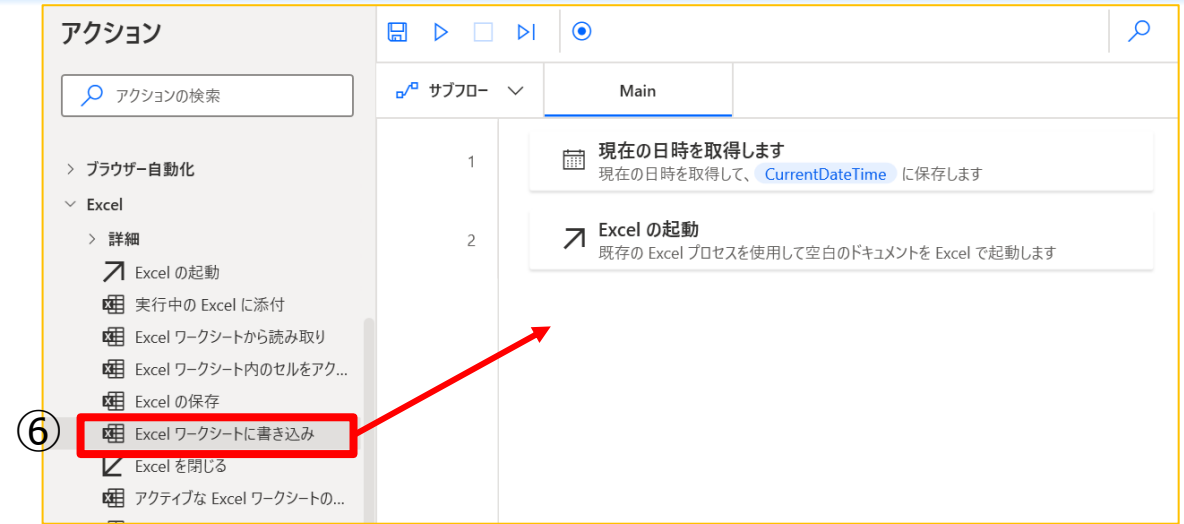
⑤保存

変数が自動で生成されます



① [演習 1]変数を理解する

⑥ Excel→Excelワークシートに書き込み



⑦ {x}をクリックし、CurrentDateTimeを選択

変数は%で囲まれます



① [演習 1]変数を理解する

⑧書き込むセルを指定

*列は「A」または「1」

Excel ワークシートに書き込み

Excel インスタンスのセルまたはセル範囲に値を書き込みます [詳細](#)

パラメーターの選択

全般

Excel インスタンス: %ExcelInstance%

書き込む値: %CurrentDateTime%

書き込みモード: 指定したセル上

列: A

行: 1

エラー発生時

保存 キャンセル

⑨実行ボタンまたはF5キーで実行してみましょう

アクション

アクションの検索

サブフロー Main

1 現在の日時を取得します
現在の日時を取得して、CurrentDateTime を保存します

2 Excel の起動
既存の Excel プロセスを使用して空白のドキュメントを Excel で起動します

3 Excel ワークシートに書き込み
Excel インスタンス ExcelInstance の列 'A' および行 1 のセルに値 CurrentDateTime を書き込み

① [演習 1]変数を理解する

The image shows a workflow editor on the left and an Excel spreadsheet on the right. The workflow editor has three steps:

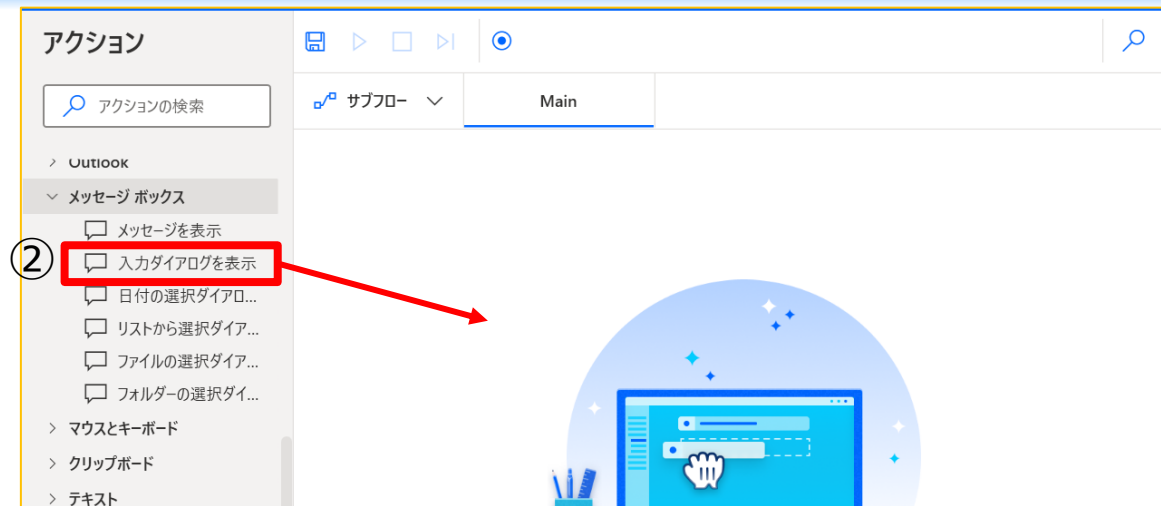
- 1 現在の日時を取得します
現在の日時を取得して、`CurrentDateTime` に保
- 2 Excel の起動
既存の Excel プロセスを使用して空白のドキュメントを
- 3 Excel ワークシートに書き込み
Excel インスタンス `ExcelInstance` の列 'A' および
`CurrentDateTime` を書き込み

The Excel spreadsheet shows the result of the workflow. The formula bar displays `2022/4/14 11:01:43`. The cell A1 contains the date and time `2022/4/14 11:01`.

A red speech bubble points to cell A1 with the text: 現時点の日付が指定したセルに入力されます

① [演習 1 -2]変数を理解する

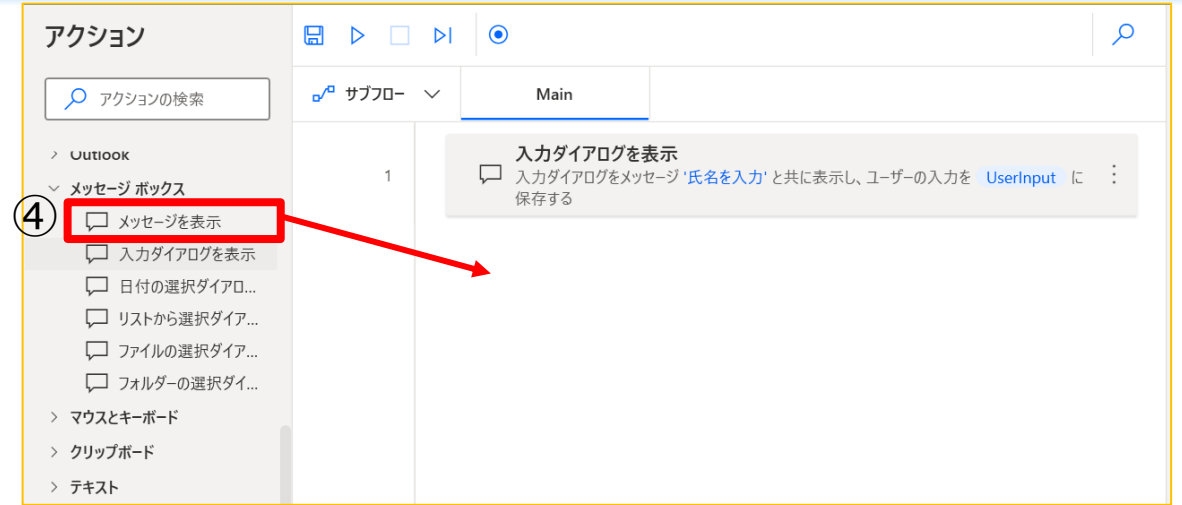
- ① +新しいフローより新規作成画面を開きます
- ②メッセージボックス→入力ダイアログを表示をワークスペースにドラッグorダブルクリック
- ③入力ダイアログ、入力ダイアログメッセージにそれぞれ入力して保存



関数が自動で生成されます
UserInput | 入力した内容
ButtonPressed | 押下したボタン

① [演習 1 -2]変数を理解する

④メッセージを表示をワークスペースにドラッグorダブルクリック



⑤メッセージボックスのタイトルを入力

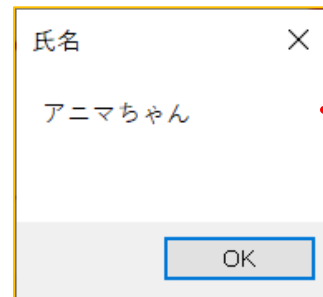
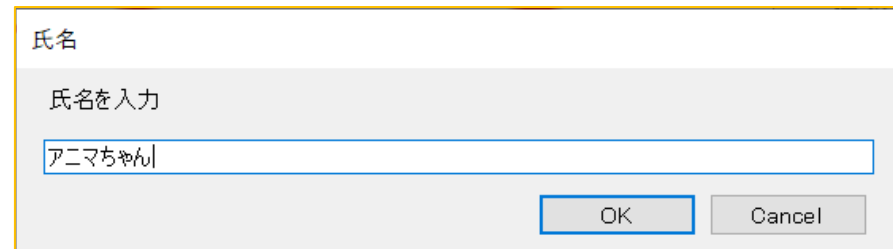
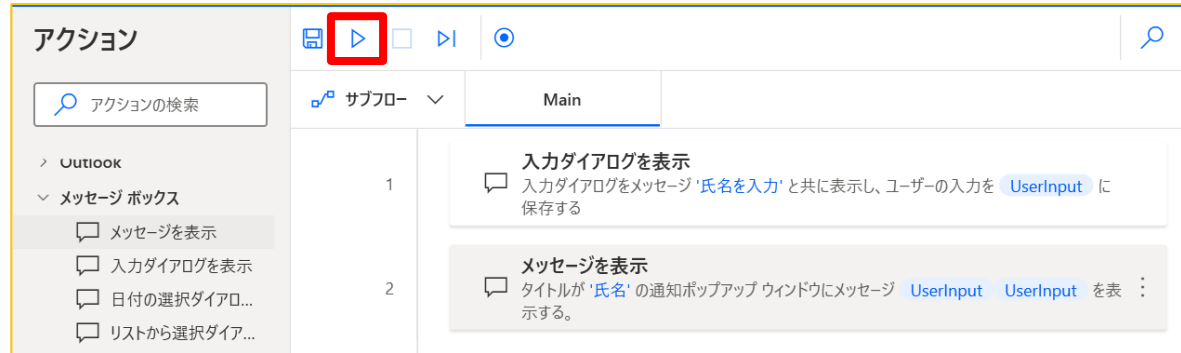
⑥{x}→UserInputをダブルクリック



⑦保存

① [演習 1 -3]変数を理解する

再生ボタンを押して作成したフローを実行することができます



入力した内容が表示されます

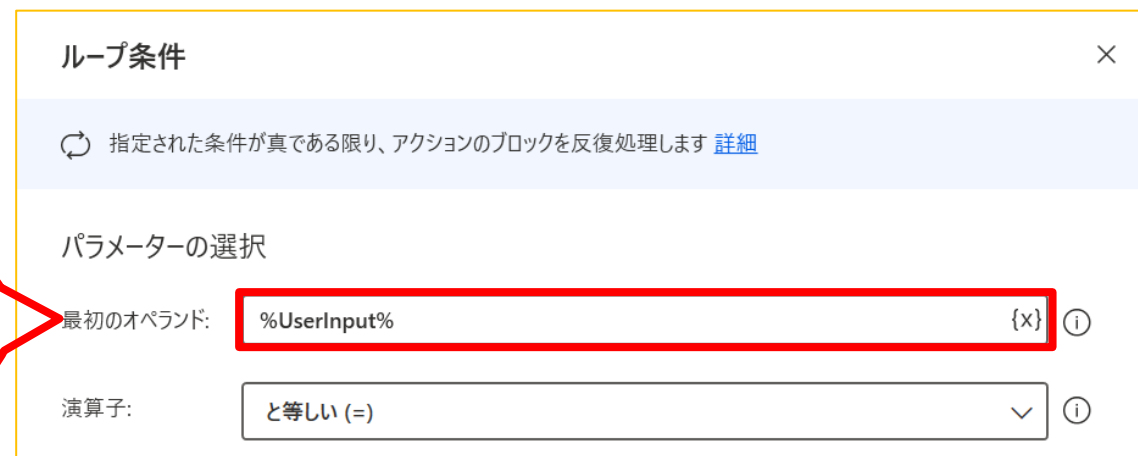
① 変数

Power Automate for Desktopでは
アクションを追加すると自動で変数が
設定されます



変数を呼び出すとき「%」で囲むこと
で変数と表現するルールです

%UserInput%
と「%」がでてきたら
変数だと思えばOK



① 変数の種類

変数には数字や文字列など様々な種類があります。自動で設定されるため、あまり意識することはありません。しかし、数字と文字の足し算ができないエラーになるなど、RPAの最初の壁となるため十分理解しておく必要があります。

データ型の種類	説明
数値型	算術演算ができる唯一の型
テキスト値型	文字列（日本語、英語、記号といった区別がなく、すべてがテキスト値型） 「%〇%」または「%"〇"%」と入力することで〇を テキスト型に指定 できます 「%" %」または「%" "" %」と入力することで 空白（値なし） を意味します
Datetime型	日付（アメリカで用いられる表記法のため、日本の日付に変更する必要あり）
ブール値型	条件に対し、Yes/NoやTrue/Falseといった2つの状態を表す
リスト型	複数の値を1つの変数に格納できる（Excelの1列目だけのイメージ）
データテーブル型	リスト型の2列以上バージョン（2列以上のExcelのイメージ）
インスタンス型	操作対象を指定するときに利用

① プロパティが存在する変数の種類

プロパティとは、そのデータ型がもっている情報のことです。

データ型の種類	説明
ファイル型	取得したファイルの情報が格納された型
フォルダ型	取得したフォルダの情報が格納された型

ファイルやフォルダの「保存場所」「フォルダ名」「作成日」を指定して利用することが可能です。

変数名の後にプロパティ名を入力することで利用できます。

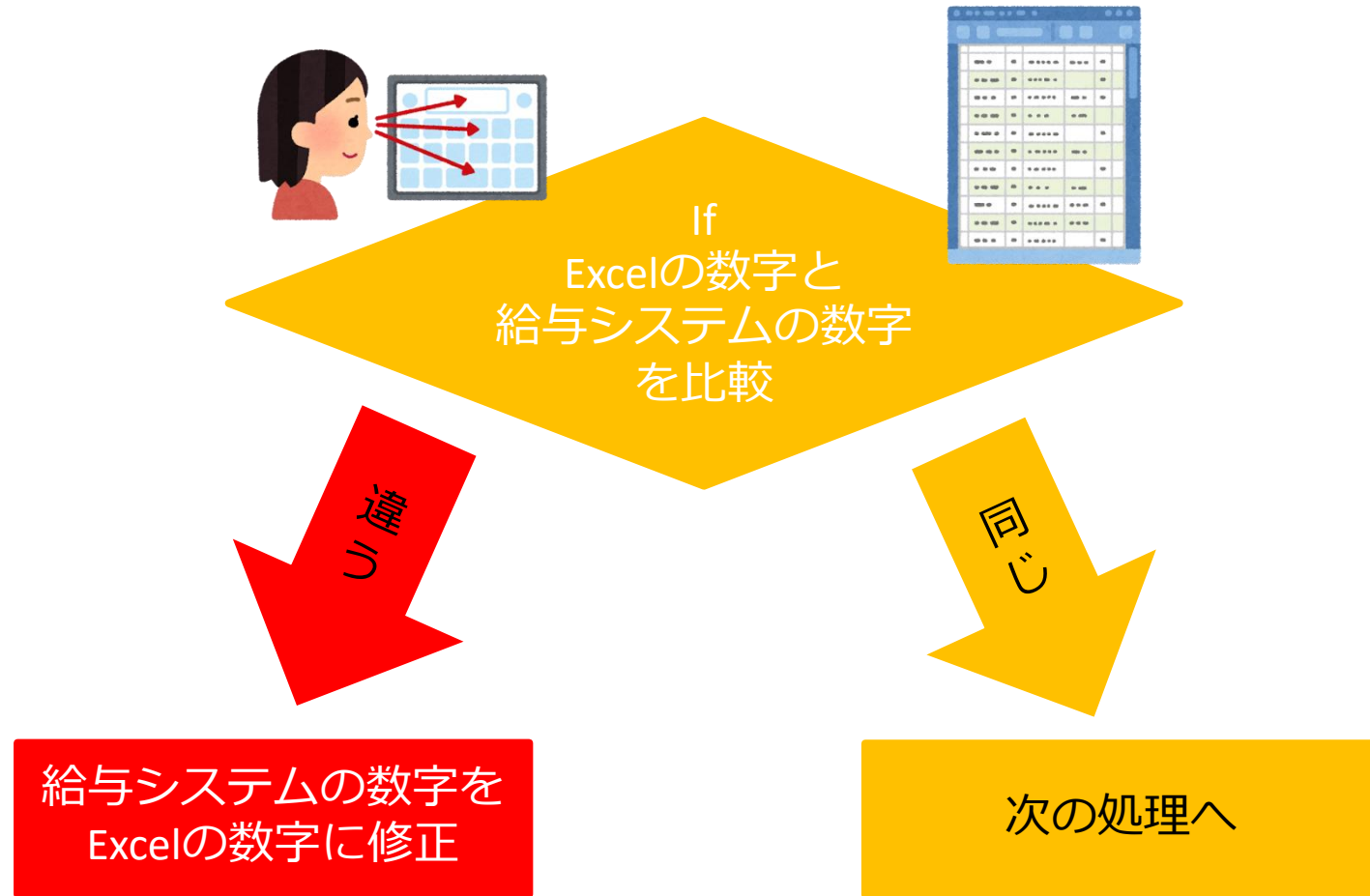
変数名.Name (フォルダ名)

変数名.Parent (フォルダの格納場所)

変数名.CreationTime (作成日)

① 条件分岐

条件に応じて次に行うアクションをつかい分けたいときに利用します。
数字の確認・修正作業など、よく利用するアクションですのでしっかり覚えましょう。



① 「If」 アクション

変数の値が指定した条件に合致したか判定し、次のアクションを選択できるようになります。オペランドには変数や指定した数値等を入力します。

条件にしたい変数や定数

比較したい変数や定数

If

このステートメントで指定した条件を満たす場合に実行する、アクションのブロックの開始を示します [詳細](#)

パラメーターの選択

最初のオペランド:

演算子:

2 番目のオペランド:

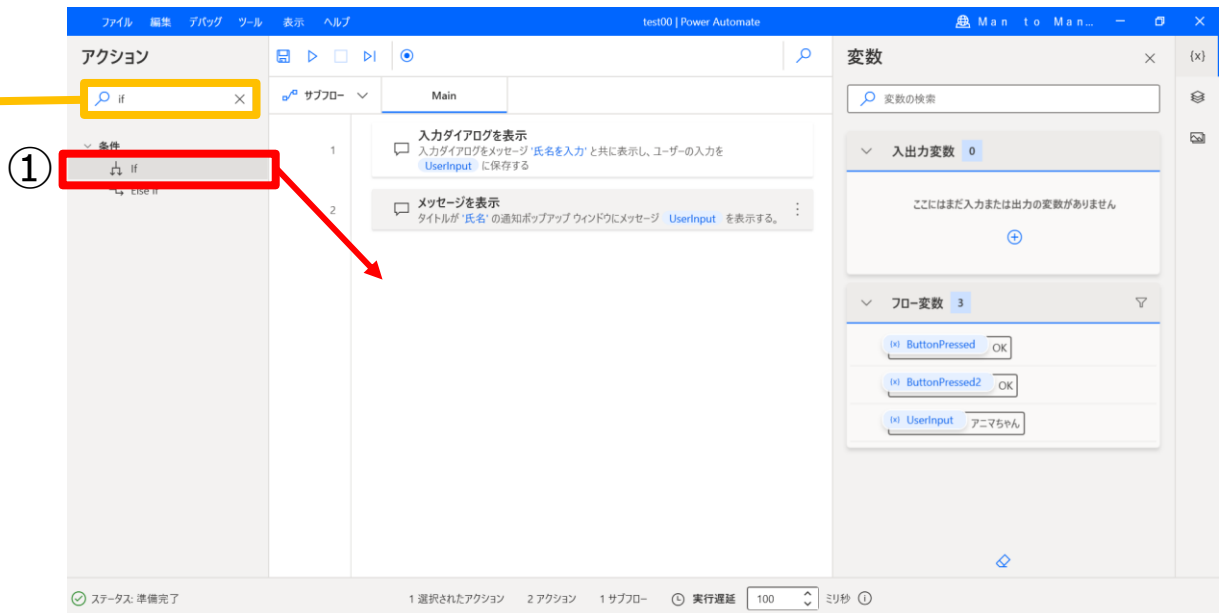
① [演習2]条件分岐を理解する

演習1で作成したフローに追加して試してみましよう。

ダイアログに入力がなかったときに、再入力するフローに修正します。

① ifをワークフローに追加

キーワードで検索
することも可能



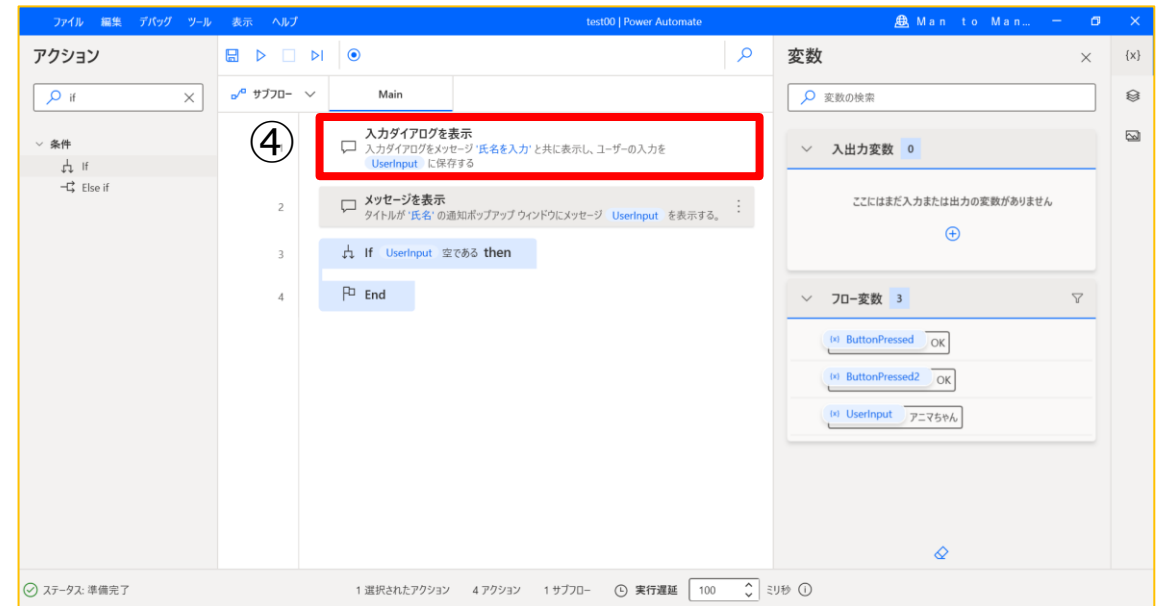
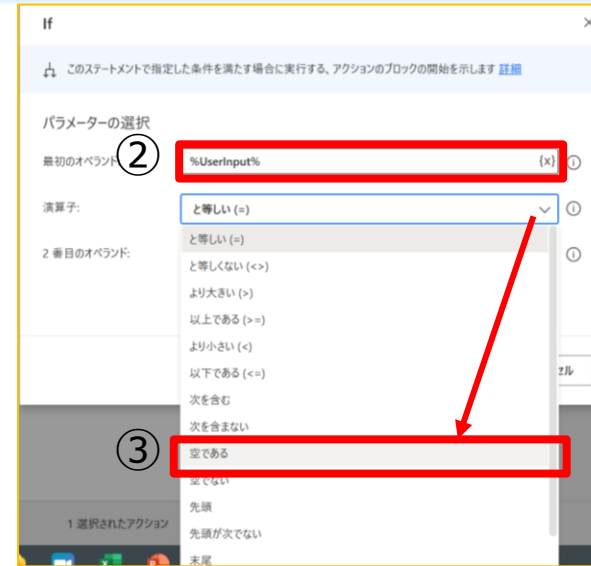
① [演習2]条件分岐を理解する

②最初のオペランドに%UserInput%を選択

③演算子「空である」を選択

④入力ダイアログを表示を「Ctrl+C」または
右クリック→コピー

*コピーの操作には少し待ち時間がでます



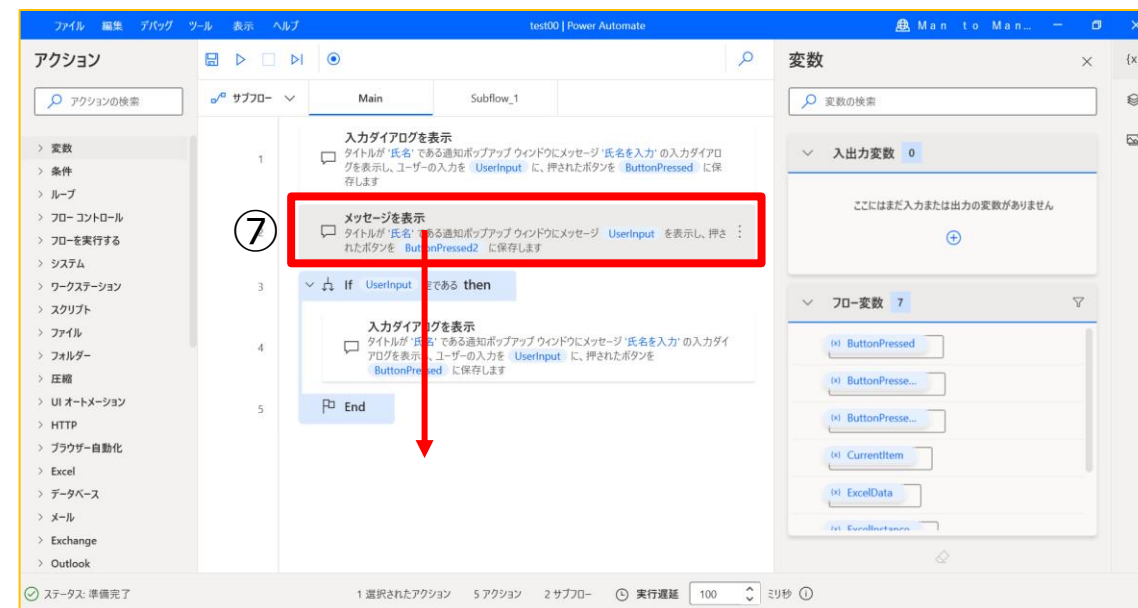
④ [演習2]条件分岐を理解する

⑤ 「Ctrl+V」 または右クリック→貼り付け

⑥ 入力ダイアログを表示をifとEndの間にドラッグアンドドロップして移動します



⑦ メッセージを表示をEndの下にドラッグ



① [演習2]条件分岐を理解する

再生ボタンを押して作成したフローが正常に動作するか確認しましょう



氏名

氏名を入力

OK Cancel



氏名

氏名を入力

OK Cancel



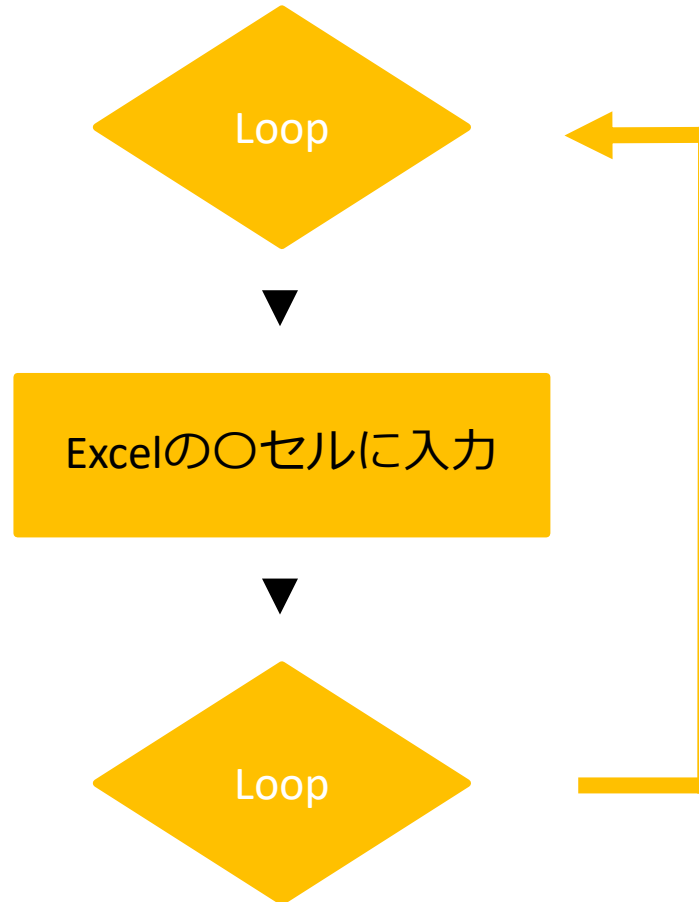
氏名

アニマちゃん

OK

① 繰り返し

アクションを繰り返し操作したいときに利用します。
データの入力や目視確認、回数や条件を指定して繰り返し作業を行いたいときに利用します。



① 「Loop」 アクション

LoopからEndまで**指定した回数分**繰り返すアクションです。

右図の場合、1から5まで5回繰り返します。

Loop

指定された数のアクションのブロックを反復処理します [詳細](#)

パラメーターの選択

開始値: {x} ⓘ

終了: {x} ⓘ

増分: {x} ⓘ

> 生成された変数 [LoopIndex](#)

① 「For each」アクション

データテーブル型やリスト型といった複数の要素を持つ変数のみに使用可能なアクションです。

データの行数分繰り返しが行えます。

The screenshot shows a configuration window titled "For each" with a close button (X) in the top right corner. Below the title bar, there is a light blue header area containing a refresh icon and the text: "リスト、データテーブル、またはデータ行にあるアイテムを反復処理して、アクションのブロックを繰り返して実行します" followed by a link labeled "詳細". Below this is the section "パラメーターの選択". Under "反復処理を行う値:", there is a text input field with a blue border, containing a vertical bar and a placeholder "{x}", with an information icon (i) to its right. Under "保存先:", there is a blue pill-shaped button labeled "CurrentItem" followed by a placeholder "{x}". At the bottom right, there are two buttons: a blue "保存" button and a white "キャンセル" button with a grey border.

* 単一の値のみの変数には利用できません

① 「Loop条件」アクション

最初のオペラントと2番目のオペラントの条件が一致している場合に、繰り返しを行うアクションです。

右図の場合、%UserInput%が「アニメ」となっているかぎり繰り返し続けます。

ループ条件

指定された条件が真である限り、アクションのブロックを反復処理します [詳細](#)

パラメーターの選択

最初のオペラント: {x} ⓘ

演算子: ▼ ⓘ

2番目のオペラント: {x} ⓘ

Office STOK

Excel編



① VBAとPower Automate for Desktop

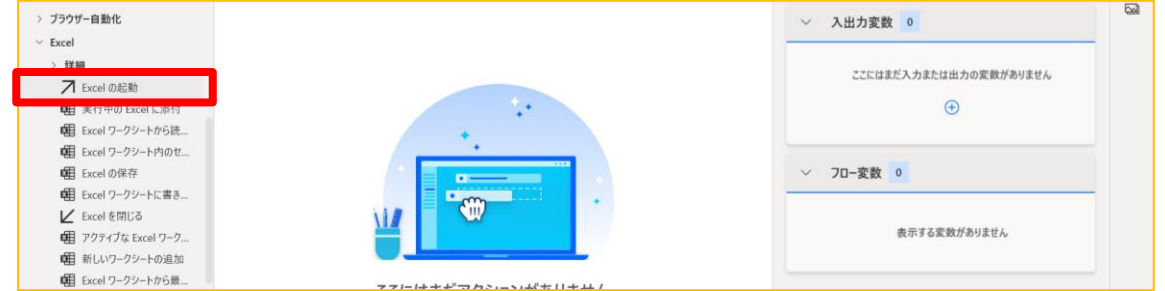
自動化する点ではVBAと共通していますが、Power Automate for Desktopではプログラミングの知識がなくても業務の自動化が可能です。

Excel操作のためのアクションがいくつか用意されています。



① Excelを開く

Excel→Excelの起動をワークフローへ
ドラッグ



次のドキュメントを開く
開きたいExcelを選択

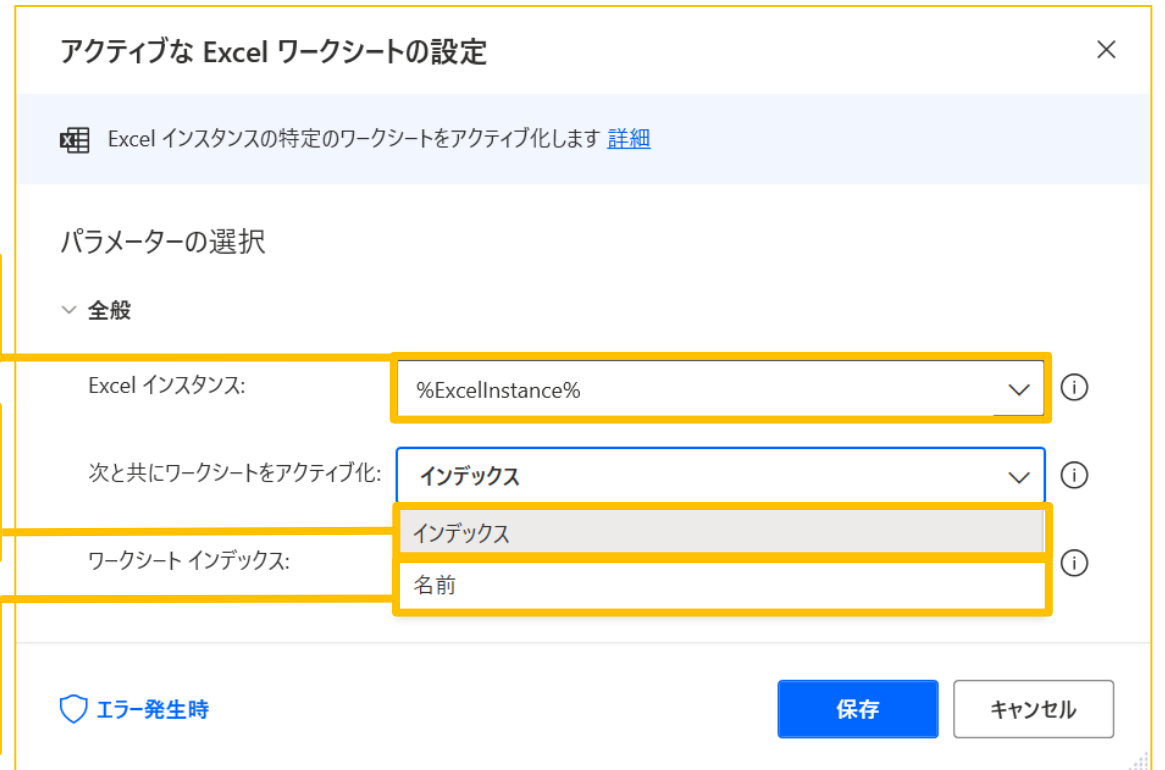
*パスとはファイルの保存位置を指定
する文字列のこと

自動で変数が生成されます



① Excelワークシートを選択する

Excelを開いたときに選択されているワークシートでその後処理を実行するため、複数のワークシートがある場合、指定する必要があります



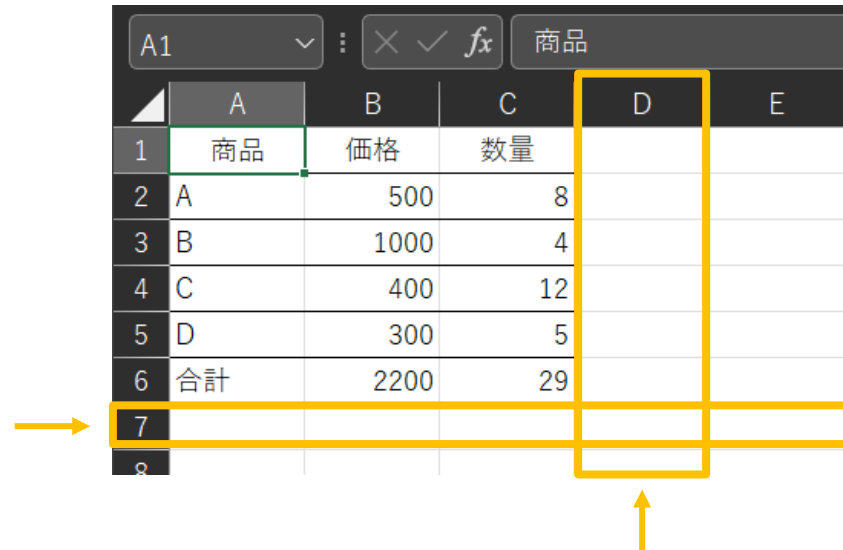
開きたいExcelの変数を選択

シートの番号で指定する場合
左から準備 1.2.3...
シートを増やすと番号がずれる

シート名称で指定する場合
シートが増やしても対応できるが、
シート名が変更されるとだめ

① データの抽出 方法①

選択したワークシートから数値や文字列を取得する方法を学びます
最初の「空の列」と「空の行」を取得することで、データの範囲を特定します



The image shows an Excel spreadsheet with a table of data. The table has columns A, B, C, D, and E, and rows 1 through 8. The data is as follows:

	A	B	C	D	E
1	商品	価格	数量		
2	A	500	8		
3	B	1000	4		
4	C	400	12		
5	D	300	5		
6	合計	2200	29		
7					
8					

A yellow box highlights the range from row 7 to row 8 and column D to column E. A yellow arrow points to the start of row 7, and another yellow arrow points to the start of column D. The formula bar at the top shows 'A1' and '商品'.

① データの抽出（空白行・列の取得）

方法①

Excel→Excelワークシートから最初の空の列や行を取得の起動をワークフローへドラッグ



Columnは**列**
Rowは**行**
のことです

開きたいExcelの変数を選択

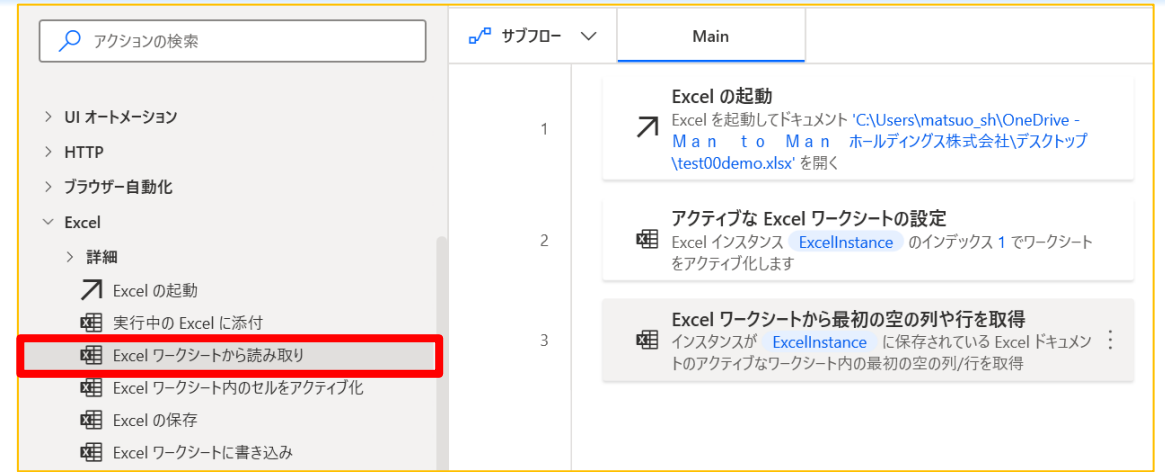
最初の空の**列**数が格納されます

最初の空の**行**数が格納されます
D列の場合「4」となります



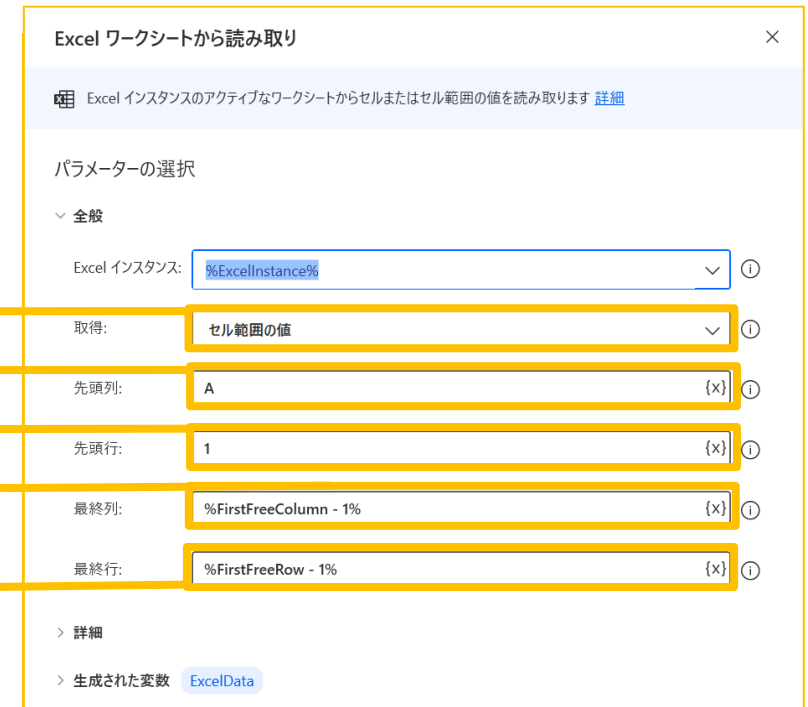
① データの抽出（データの読み取り） 方法①

Excel→Excelワークシートから読取
をワークフローヘドラッグ



A 1 セルから空の行列のひとつ前までの
データを範囲指定する方法です

- セルの範囲の値を選択
- 「A」または「1」
- 「1」
- 変数%FirstFreeColumn%を選択し、
後ろの%の前に「-1」を入力
- 変数%FirstFreeRow%を選択し、
後ろの%の前に「-1」を入力



① データの抽出（データの読み取り）

方法①

Excelの1行目が列名となっている場合
後でテーブル内のデータを検索する際に
列名で指定できるため、オンにしておき
ます

Excel ワークシートから読み取り

☑ Excel インスタンスのアクティブなワークシートからセルまたはセル範囲の値を読み取ります [詳細](#)

Excel インスタンス: %ExcelInstance% (i)

取得: セル範囲の値 (i)

先頭列: A {x} (i)

先頭行: 1 {x} (i)

最終列: %FirstFreeColumn - 1% {x} (i)

最終行: %FirstFreeRow - 1% {x} (i)

▼ 詳細

セルの内容をテキストとして取得: (i)

範囲の最初の行に列名が含まれています: (i)

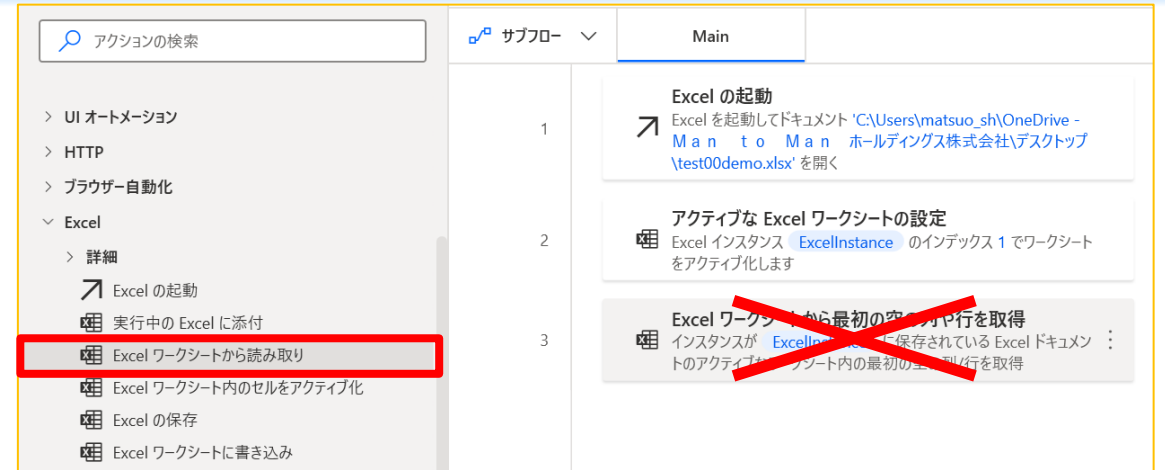
> 生成された変数 [ExcelData](#)

エラー発生時

保存 キャンセル

① データの抽出（データの読み取り） 方法②

Excel→Excelワークシートから読取
をワークフローヘドラッグ



空の列や行を取得
しない方法もあります

ワークシートに含まれる使
用可能なすべての値



① データ抽出（データの確認）

フローを実行すると変数に値が格納され
該当の変数をダブルクリックすると
内容を確認することができます

変数の値

ExcelData (Datatable)

#	Column1	Column2	Column3
0	商品	価格	数量
1	A	500	8
2	B	1000	4
3	C	400	12
4	D	300	5
5	合計	2200	29

閉じる

変数

変数の検索

入出力変数 0

ここにはまだ入力または出力の変数がありません

+

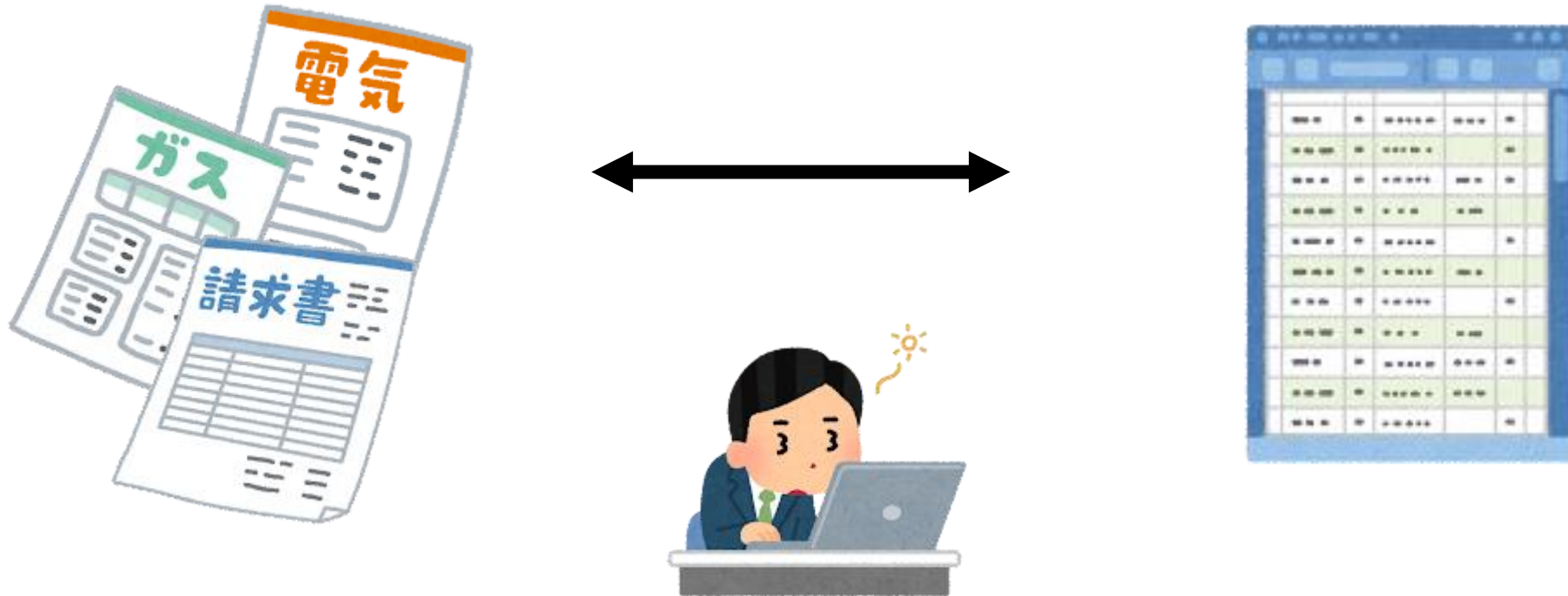
フロー変数 5

- (x) ButtonPressed OK
- (x) ExcelData 行, 3 列
- (x) ExcellInstance Excel インスタンス
- (x) FirstFreeColumn 4
- (x) FirstFreeRow 7

① ExcelからExcelへのデータ書き込み

データを抽出できたら、そのデータを他のExcelへ入力する方法を学びます。変数に格納されたデータを用いて他のExcelへデータ書き込みができるため、手作業でのExcel間を行ったり来たりする手間がかかりません。

Excelからデータを取得し、条件に合ったデータを別のExcelへ入力するフローを作成してみましょう。



① ExcelからExcelへのデータ書き込み

①読み込んだExcelDataの値を新しいExcelに貼り付ける

`%ExcelData%`

②読み込んだExcelDataの1行目の値を新しいExcelに貼り付ける

`%ExcelData[0]%`

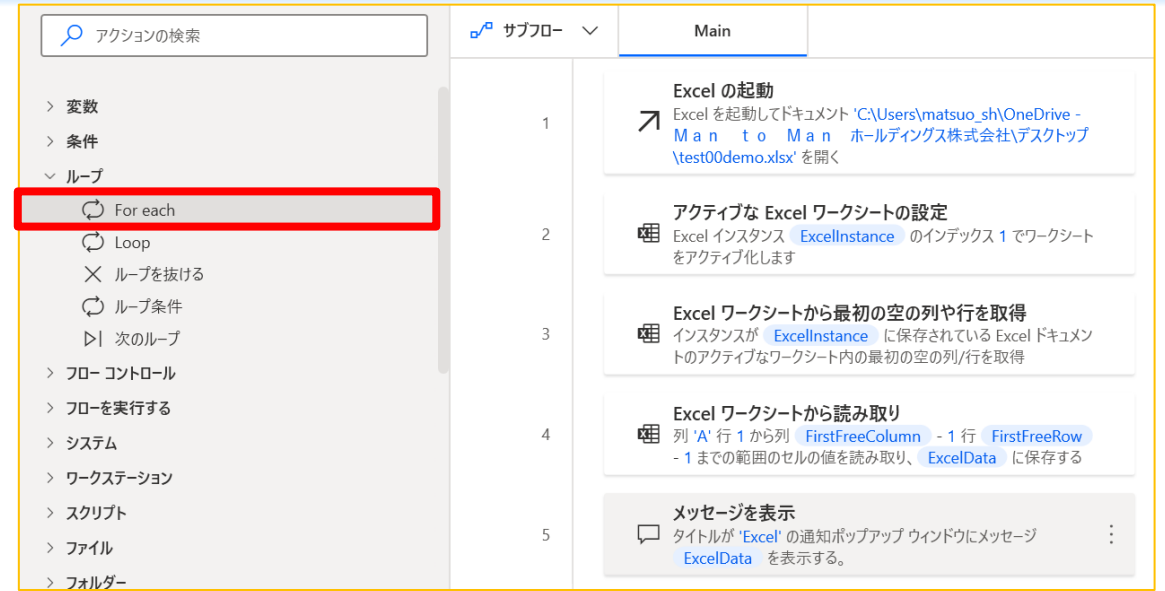
③読み込んだExcelDataの2行目・3列目の値を新しいExcelに貼り付ける

`%ExcelData[1][2]%` または `%ExcelData[1]['数量']%`

① データテーブルの行数分ループ処理を行う

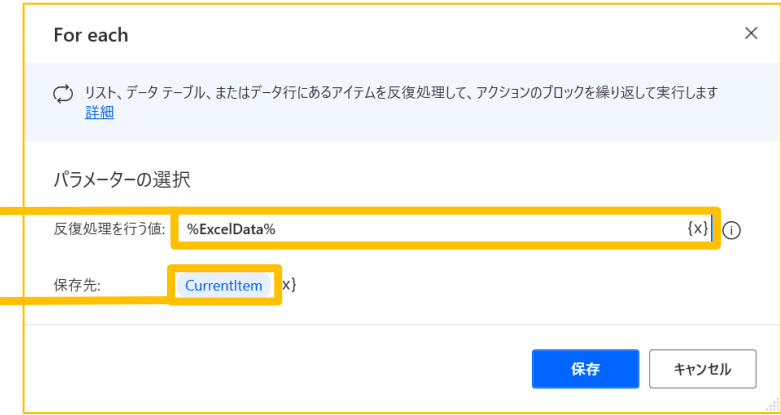
データテーブル上の数値条件によって処理をかえるフローを作成します

まず、1行ずつ変数に格納する必要があるため、ループ処理 (For each) を使います



項目数分繰り返したい
データテーブルが格納
された変数を選択

ループ処理の中で
現在選択中の項目が
格納されます



CurrentItem
には何が格納
されるか確認

① 条件に合うデータを抽出する（なくてもOK）

数量が10以上の商品データを抽出する
フローを作成してみましょう

数値の条件を行う場合は、値のデータ型
が異なる場合、エラーとなります

そのため、まずは値を数値型に変換する
処理を行います

テキスト→テキストを数値に変換を
ドロップ

%CurrentItem%を選択し、
後ろの%の前に「数量」を入力することで
数量の列にある値を指定できます

The screenshot shows the Power Automate flow editor. On the left, the 'Text' task pane is open, and the 'Text to Number' action is highlighted with a red box. A red arrow points from this box to the 'For each' loop in the flow. The flow steps include: 'Excel を起動してドキュメント C:\Users\matsuo_sn\OneDrive\... n ホールディングス株式会社\デスクトップ\test00demo.xls', 'アクティブな Excel ワークシートの設定', 'Excel ワークシートから最初の空の列や行を取得', 'Excel ワークシートから読み取り', 'メッセージを表示', 'For each CurrentItem in ExcelData', and 'End'.

The screenshot shows the configuration dialog for the 'Text to Number' action. The title is 'テキストを数値に変換'. The description says '数値のテキスト表現を、数値を含む変数に変換します 詳細'. Under 'パラメーターの選択', the '全般' section is expanded. The '変換するテキスト:' field contains '%CurrentItem[数量]%' and is highlighted with a yellow box. Below it, the '生成された変数' section shows 'TextAsNumber'. At the bottom, there are buttons for 'エラー発生時', '保存', and 'キャンセル'.

① 条件を設定する

条件→ifをドラッグ

The screenshot shows a workflow editor interface. On the left, a sidebar lists various actions, with 'If' highlighted in a red box. A red arrow points from this box to the 'If' condition being added to a workflow step in the main editor. The workflow step is titled 'For each CurrentItem in ExcelData' and contains a sub-step 'テキストを数値に変換' (Convert text to number) with parameters 'CurrentItem' and 'TextAsNumber'.

%TextAsNumber%を選択し
条件を設定

The 'If' configuration dialog box is shown. It has a title bar 'If' and a close button. Below the title, there is a description: 'このステートメントで指定した条件を満たす場合に実行する、アクションのブロックの開始を示します 詳細'. Underneath, there is a section 'パラメーターの選択' (Parameter selection). It contains three input fields: '最初のオペランド:' (First operand) with the value '%TextAsNumber%', '演算子:' (Operator) with the value '以上である (>=)', and '2番目のオペランド:' (Second operand) with the value '10'. Each field has a small 'i' icon to its right. At the bottom right, there are two buttons: '保存' (Save) and 'キャンセル' (Cancel).

① 記入先のExcelを開く

ファイルは一度開けばよいいため、
For eachの上に配置します

The screenshot shows a Power Automate flow with the following steps:

- 4. 列 'A' 行 1 から列 FirstFreeColumn - 1 行 FirstFreeRow - 1 までの範囲のセルの値を読み取り、ExcelData に保存する
- 5. メッセージを表示
タイトルが 'Excel' の通知ポップアップ ウィンドウにメッセージ ExcelData を表示する。
- 6. For each CurrentItem in ExcelData (highlighted with a red box and arrow)
- 7. テキストを数値に変換
テキスト CurrentItem [数量] を数値に変換し、TextAsNumber に保存する
- 8. If TextAsNumber >= 10 then

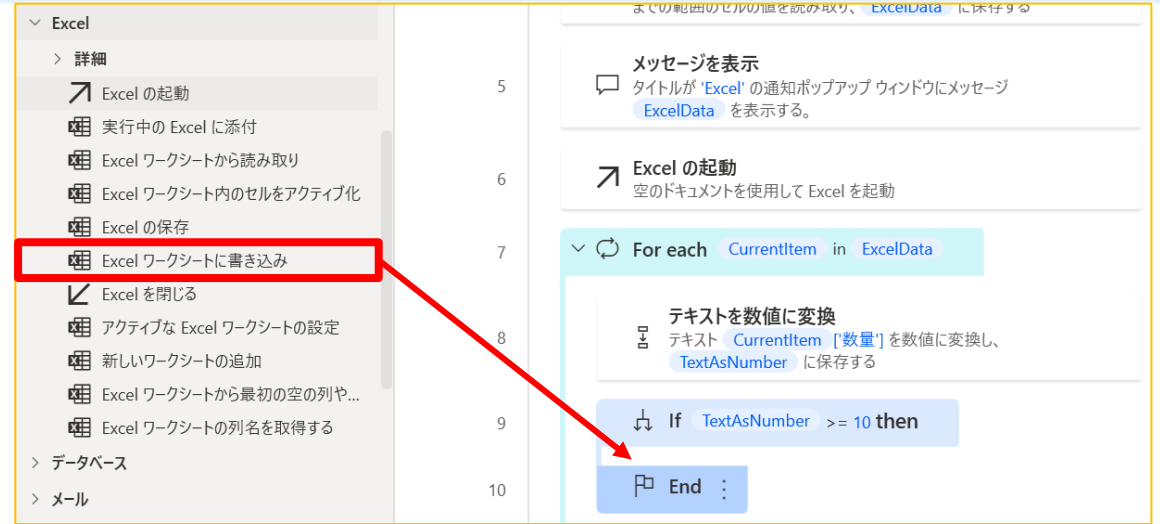
そのまま保存します

The 'Excel の起動' dialog box configuration is as follows:

- 新しい Excel インスタンスを起動するか、Excel ドキュメントを開きます [詳細](#)
- パラメーターの選択
- 全般
 - Excel の起動: 空のドキュメントを使用
 - インスタンスを表示する:
- 詳細
- 生成された変数: ExcelInstance2
- エラー発生時
- 保存 (highlighted in red)
- キャンセル

① 指定したセルに値を書き込む

Excel→Excelワークシートに書き込みをドラッグ



書き込まれるExcel

%CurrentItem%

「A」または「1」

「1」

Excel ワークシートに書き込み

Excel インスタンスのセルまたはセル範囲に値を書き込みます [詳細](#)

パラメーターの選択

全般

Excel インスタンス: %ExcelInstance2% (x) ⓘ

書き込む値: %CurrentItem% (x) ⓘ

書き込みモード: 指定したセル上 (x) ⓘ

列: A (x) ⓘ

行: 1 (x) ⓘ

エラー発生時

保存 キャンセル

① 指定したセルに値を書き込む

セル単位で書き込みたい場合は、右図のように指定することも可能です
[「値の取り出し方」参照](#)

1	商品	価格	数量
2	A	500	8
3	B	1000	4
4	C	400	12
5	D	300	5
6	E	2200	29

%CurrentItem['商品']%

%CurrentItem['価格']%

%CurrentItem['数量']%

① 変数の設定

1行目への書き込みはできましたが
行を下にずらしながら書き込むために
まず行番号を変数に置き換える必要が
あります

変数→変数の設定をFor eachの上に
配置します

The screenshot shows the Power Automate interface with a 'For each' loop. The left sidebar lists variable actions, with '{x} 変数の設定' highlighted in red. A red arrow points from this action to its configuration in the main workspace. The configuration shows the variable 'CurrentItem' from 'ExcelData' being used in a 'Text to number' action.

初期値「1」を入力

The 'Variable Settings' dialog box is shown. It contains the following text: '{x} 新規や既存の変数に値を設定する、新しい変数を作成する、または以前作成した変数を上書きする [詳細](#)'. Below this, the '設定:' field is set to 'NewVar {x}'. The '宛先:' field contains the value '1' and is highlighted with a yellow box. At the bottom right, there are '保存' (Save) and 'キャンセル' (Cancel) buttons.

① 行番号を増加させる

変数→変数を大きくするをExcelワークシートに書き込みとEndの間にドラッグ

The screenshot shows the Power Automate flow editor interface. On the left, a task pane lists actions under '変数' (Variables), with '変数を大きくする' (Increase variable) highlighted by a red box. A red arrow points from this box to the flow canvas. The flow canvas shows a sequence of steps: '変数の設定' (Set variable), 'For each' loop containing 'テキストを数値に変換' (Convert text to number), an 'If' condition, and finally the 'Excel ワークシートに書き込み' (Write to Excel worksheet) action. The '変数を大きくする' action is being dragged from the task pane into the flow, positioned between the 'Excel ワークシートに書き込み' and 'End' actions.

%NewVar%を選択
下にずらしたい行数を入力

The screenshot shows the configuration dialog for the '変数を大きくする' (Increase variable) action. The dialog has a title bar and a close button. Below the title, there is a description: '変数の値を一定量大きくする 詳細'. Under the 'パラメーターの選択' (Parameter selection) section, there is a '全般' (General) sub-section. Two input fields are present: '変数名:' (Variable name) with the value '%NewVar%' and '大きくする数値:' (Increase by value) with the value '1'. Both fields have a small 'X' icon to the right. At the bottom right, there are '保存' (Save) and 'キャンセル' (Cancel) buttons.

① 行番号を変数に置き換える

▼ If TextAsNumber >= 10 then

Excel ワークシートに書き込み
Excel インスタンス ExcellInstance2 の列 'A' および行 NewVar のセル に値 CurrentItem を書き込み

↗ 変数を大きくする
変数 NewVar を 1 大きくする

End

Excelに書き込むアクションの行を
%NewBar%に変更します

Excel ワークシートに書き込み

Excel インスタンスのセルまたはセル範囲に値を書き込みます [詳細](#)

パラメーターの選択

全般

Excel インスタンス: %ExcellInstance2%

書き込む値: %CurrentItem%

書き込みモード: 指定したセル上

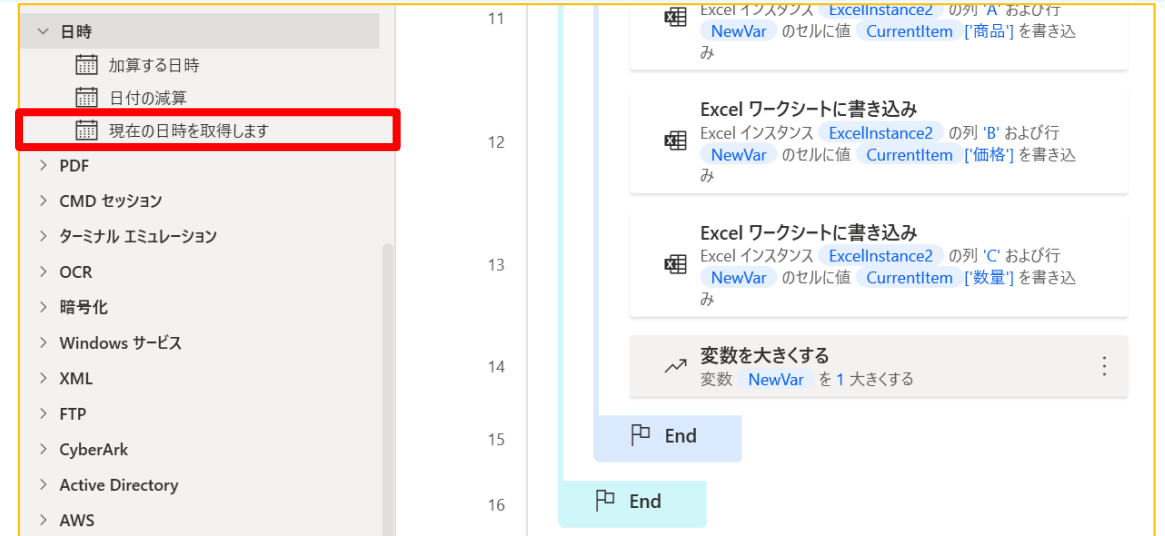
列: A

行: %NewVar%

エラー発生時 保存 キャンセル

① Excelを名前を付けて保存（現在日時を取得）

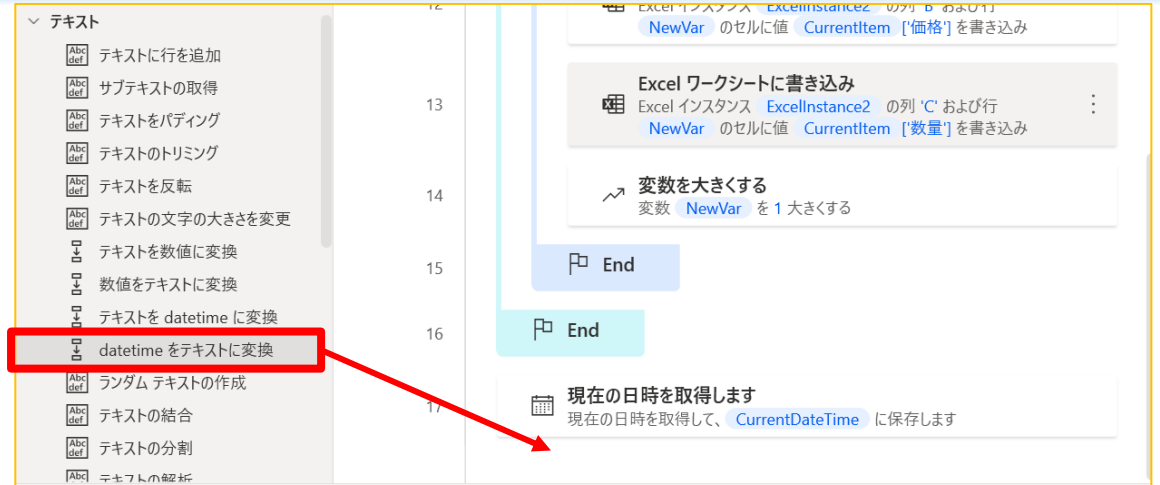
書き込んだExcelを名前を付けて保存
します
保存名称は任意でつけられます
ここでは日付を保存名に追記する方法
を紹介します
日時→現在日時を取得しますを
ドロップ



そのまま保存

① Excelを名前を付けて保存（現在日時のデータ型を変更）

日時データが格納された変数は
Datetime型のため、テキストとして
利用できません
そのためテキスト型に変更します



%CurrentDateTime%を選択
カスタムを選択

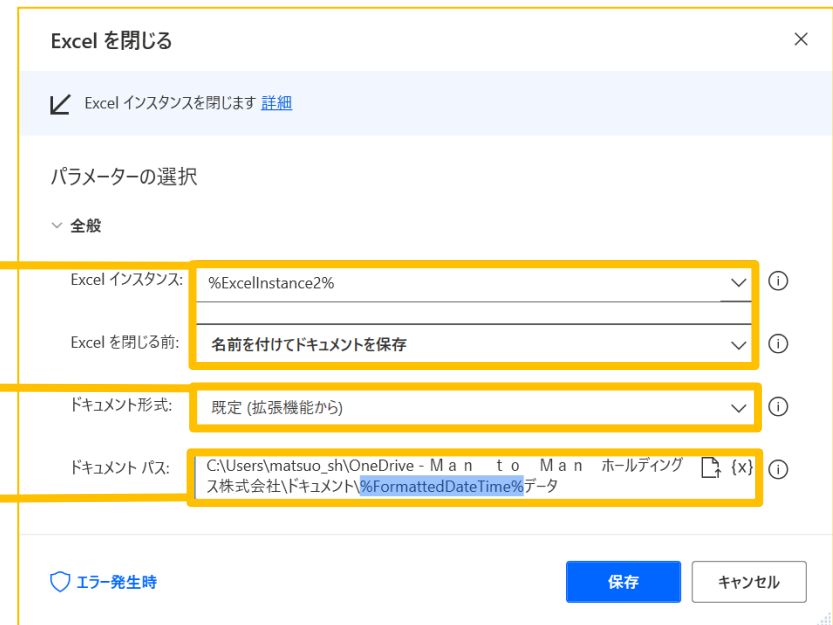
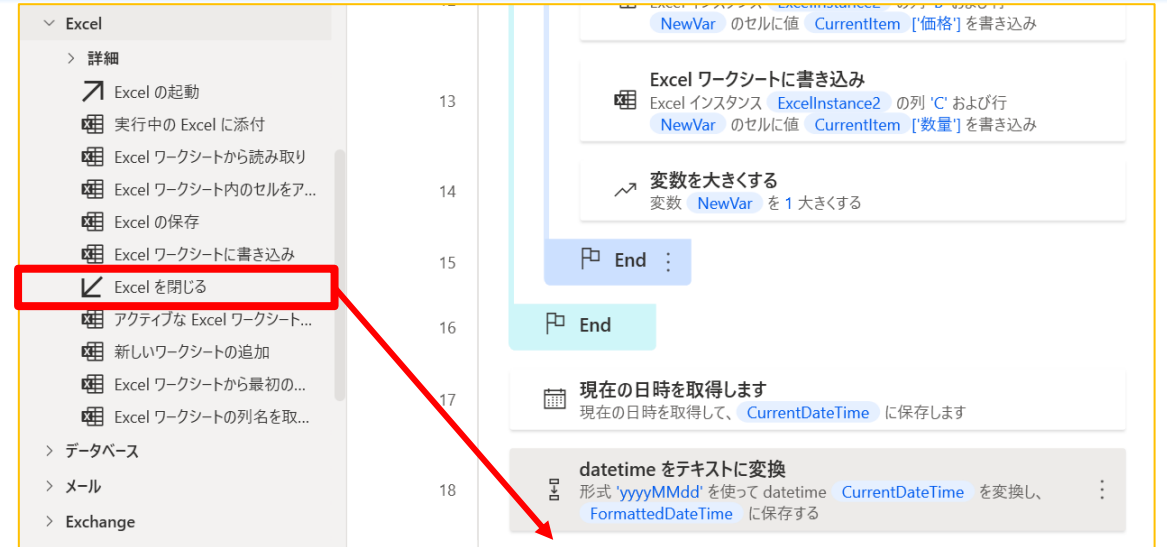
表示させたい形式を指定します
yyyy | 年 (2022)
MM | 月 (04)
dd | 日 (01)

The dialog box is titled 'datetime をテキストに変換'. It contains the following fields and options:

- 指定されたカスタム形式を使って、datetime 値をテキストに変換します [詳細](#)
- パラメーターの選択
- 変換する datetime: %CurrentDateTime% (X) ⓘ
- 使用する形式: カスタム ⓘ
- カスタム形式: yyyyMMdd (X) ⓘ
- サンプル: 20200519
- 生成された変数: FormattedDateTime
- Buttons: 保存 (Save), キャンセル (Cancel)

① Excelを名前を付けて保存（ファイル名に日付を付ける）

Excel→Excelを閉じる



%ExcelInstance2%を選択
名前を付けてドキュメントを保存を選択

現在のファイル形式のままで保存する場合
既定（拡張機能から）を選択

保存先を選択し、保存名称を入力する
入力した箇所に%FormattedDateTime%を加えること
により、日付をファイル名に加えることができる

ブラウザ編



① ブラウザの自動化の必要性

Webページから求人情報を取得したり、為替や株価を取得するなど、インターネット上の情報を取得することが可能です。

ブラウザの自動化のためのアクションがいくつか用意されています。

- ▼ ブラウザー-自動化
 - ▶ Web データ抽出
 - ▶ Web フォーム入力
 - 📄 Web ページに次が含まれる場合
 - 🕒 Web ページのコンテンツを待機します
 - 🌐 新しい Internet Explorer を起動します
 - 🌐 新しい Firefox を起動する
 - 🌐 新しい Chrome を起動する
 - 🌐 新しい Microsoft Edge を起動する
 - 📄 新しいタブを作成
 - 📄 Web ページに移動します
 - 🔗 Web ページのリンクをクリックします
 - ↓ Web ページのダウンロード リンクをクリックします
 - JS Web ページで JavaScript 関数を実行します
 - 🖱️ Web ページの要素にマウスをホバーします
 - 🔌 Web ブラウザーを閉じる

- ▼ ブラウザー-自動化
 - ▼ Web データ抽出
 - 📄 Web ページからデータを抽出する
 - 📄 Web ページ上の詳細を取得します
 - 📄 Web ページ上の要素の詳細を取得します
 - 📄 Web ページのスクリーンショットを取得します
 - ▼ Web フォーム入力
 - 📄 Web ページ上のテキスト フィールドをフォーカス
 - 📄 Web ページ内のテキスト フィールドに入力する
 - ☑️ Web ページのチェック ボックスの状態を設定します
 - 🔵 Web ページのラジオ ボタンを選択します
 - 📄 Web ページでドロップ ダウン リストの値を設定します
 - 📄 Web ページのボタンを押します

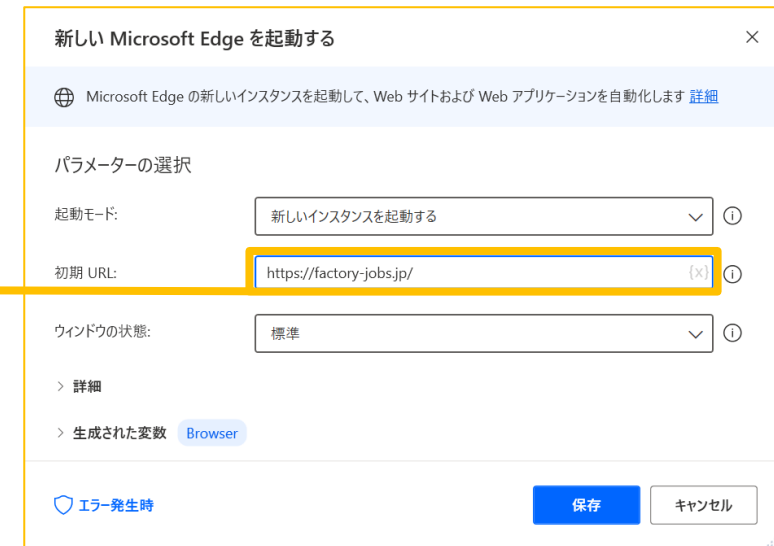
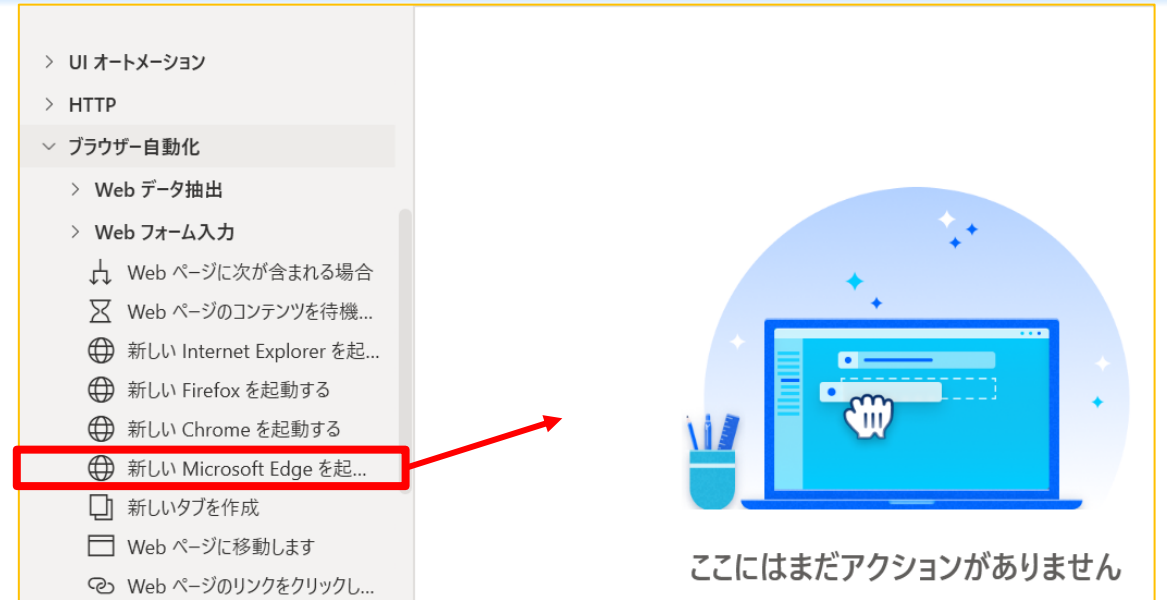
① ブラウザを開く

本講座ではMicrosoft Edgeを利用して説明します

ブラウザー自動化→新しいMicrosoft～ドラッグ

工場求人ジョブズ (<https://factory-jobs.jp/>)から求人情報を取得し、Excelに一覧を作成してみます

開きたいページのURLをコピー



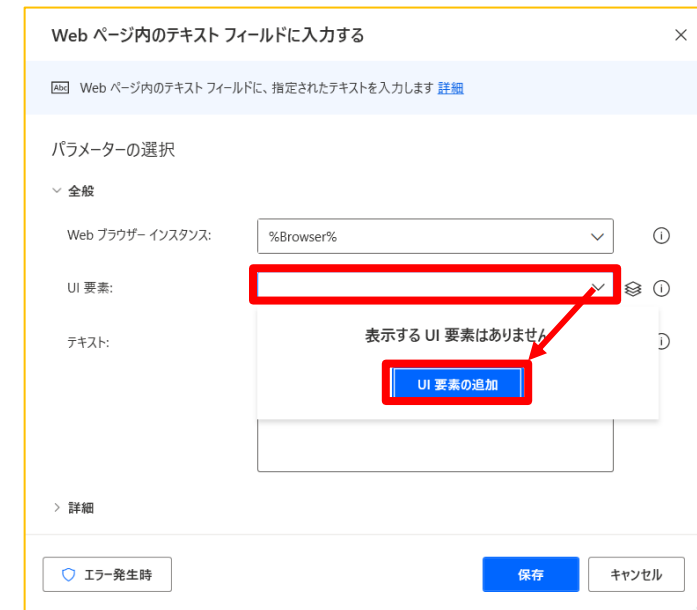
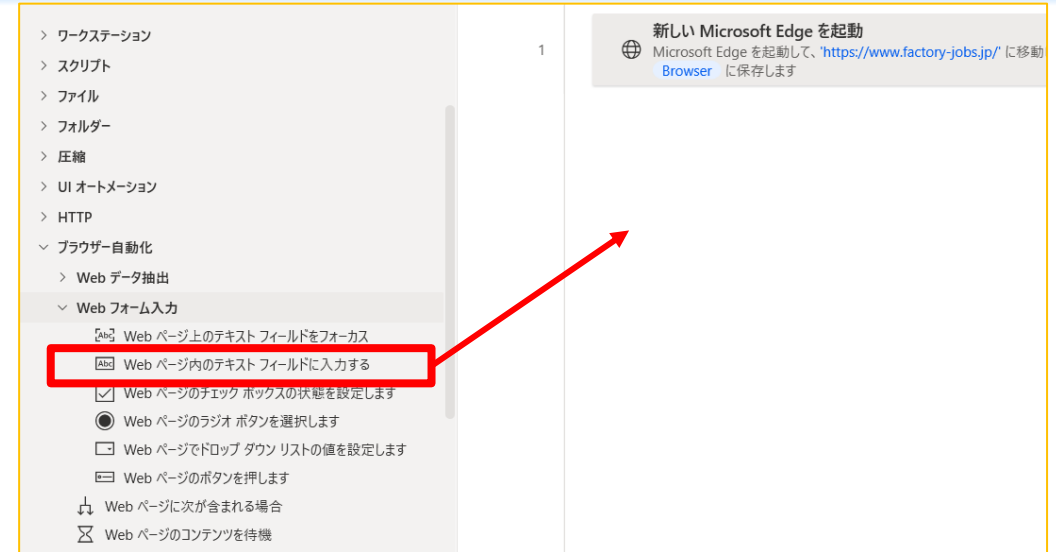
① UI 要素

操作の対象となるボタンや入力箇所を指定する必要があります

画面上に配置されたボタンや入力箇所をUI要素の部品として取り扱います

ブラウザ自動化→Webフォーム入力
→Webページ内のテキストフィールド
に入力するをドラッグ

「UI要素の追加」をクリック



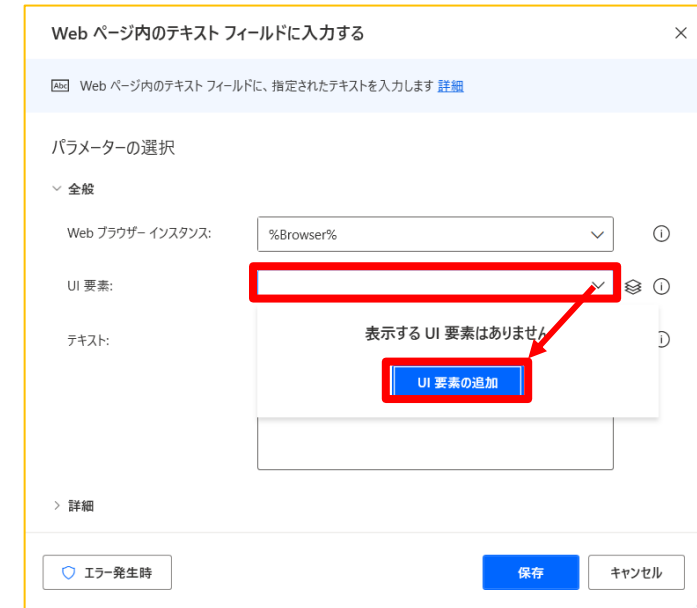
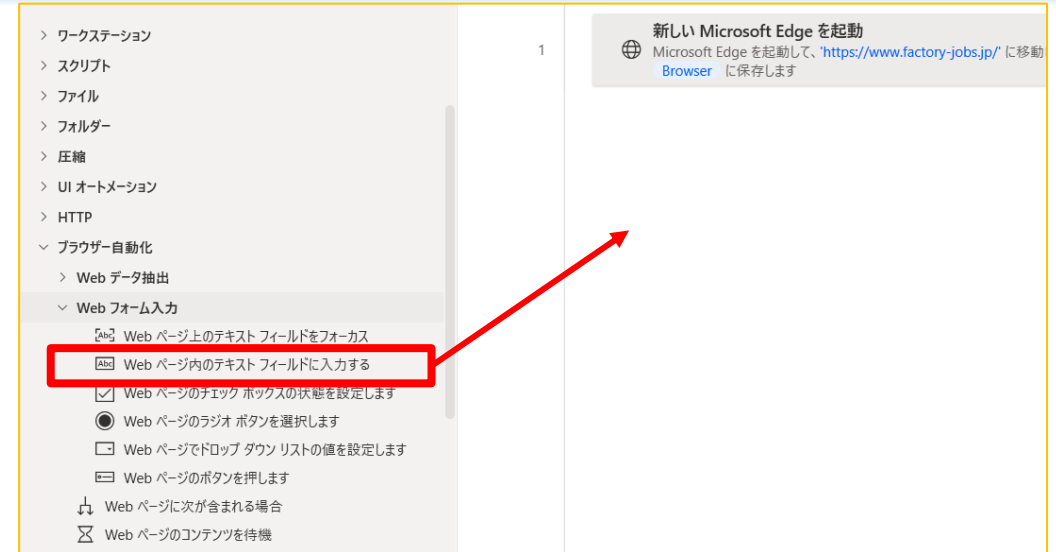
① UI 要素

操作の対象となるボタンや入力箇所を指定する必要があります

画面上に配置されたボタンや入力箇所をUI要素の部品として取り扱います

ブラウザ自動化→Webフォーム入力
→Webページ内のテキストフィールド
に入力するをドラッグ

「UI要素の追加」をクリック

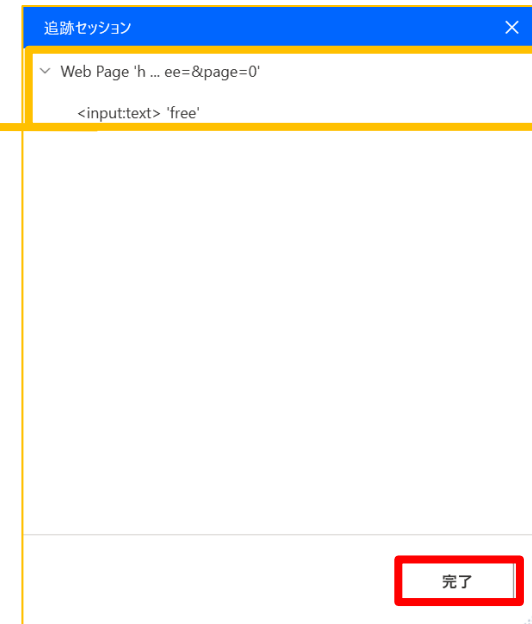


① Webページに入力する

Webページ上のUI要素にポインターを合わせると赤い枠が表示されます
Ctrlを押しながらクリック



追跡セッションに
UI要素が追加されます



UI要素を追加したら完了をクリック

① Webページに入力する

ここに入力したテキストがUI要素で
選択したフィールドに入力されます

Web ページ内のテキスト フィールドに入力する

Web ページ内のテキスト フィールドに、指定されたテキストを入力します [詳細](#)

パラメーターの選択

全般

Web ブラウザー インスタンス: %Browser%

UI 要素: Web Page 'h ... ee=&page=0' > <input.text

テキスト:

入力をエミュレートする:

> 詳細

エラー発生時

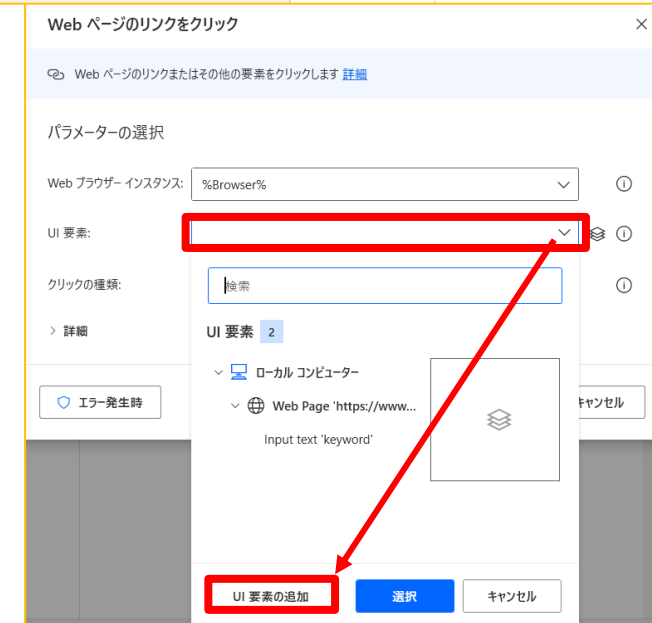
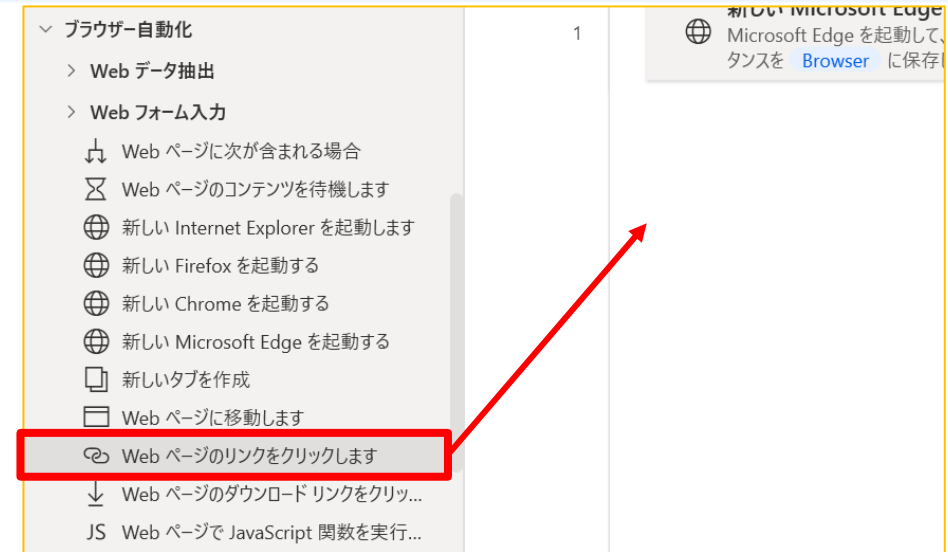
① UI 要素

操作の対象となるボタンや入力箇所を指定する必要があります

画面上に配置されたボタンや入力箇所をUI要素の部品として取り扱います

ブラウザ自動化→Webページのリンクをクリックしますをドラッグ

「UI要素の追加」をクリック

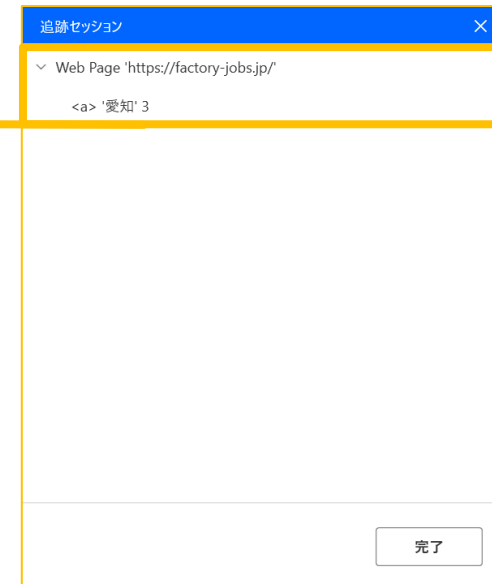


① UI要素

Webページ上のUI要素にポインターを合わせると赤い枠が表示されます
Ctrlを押しながらクリック



追跡セッションに
UI要素が追加されます



続いて勤務地を「愛知」に
設定してみましょう

チェックボックスの場合は
Webフォーム入力→
Webページ内のチェックボックスの
状態を設定します
アクションを利用してみましょう

Man to Manでのお仕事探し
求人ジョブズ

お仕事までの流れ | ブログ | カンタン登録 | 企業HP | キーワード検索

求人一覧での検索結果を表示しております

📁 職種	選択
📍 勤務地	選択
🏠 条件	選択
A キーワード	解除 未経験

Web ページのチェック ボックスの状態を設定します

Web フォームのチェック ボックスをオンまたはオフにします [詳細](#)

パラメーターの選択

全般

Web ブラウザー インスタンス: %Browser% ⓘ

UI 要素: ⓘ

チェック ボックスの状態: オン ⓘ

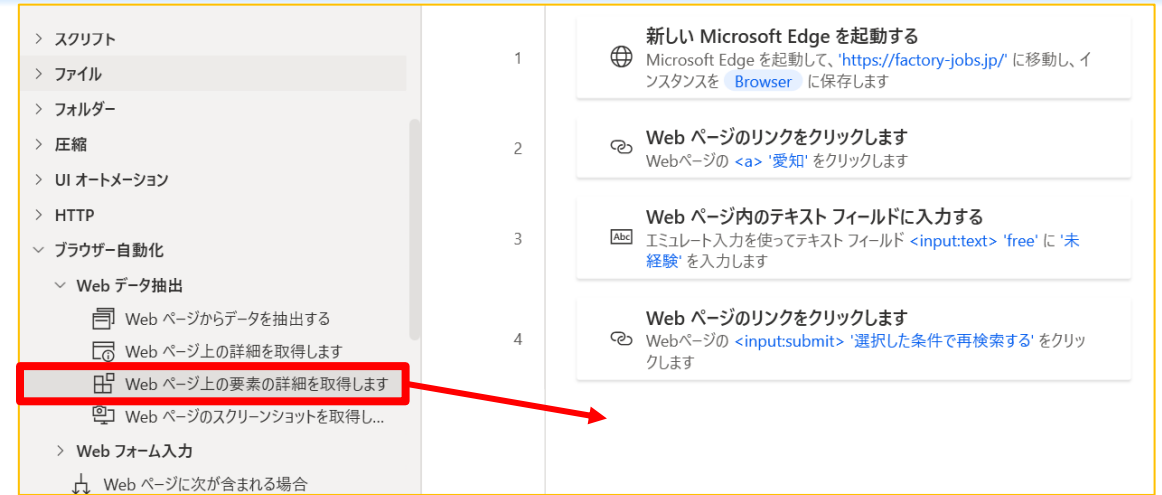
> 詳細

🛡️ エラー発生時

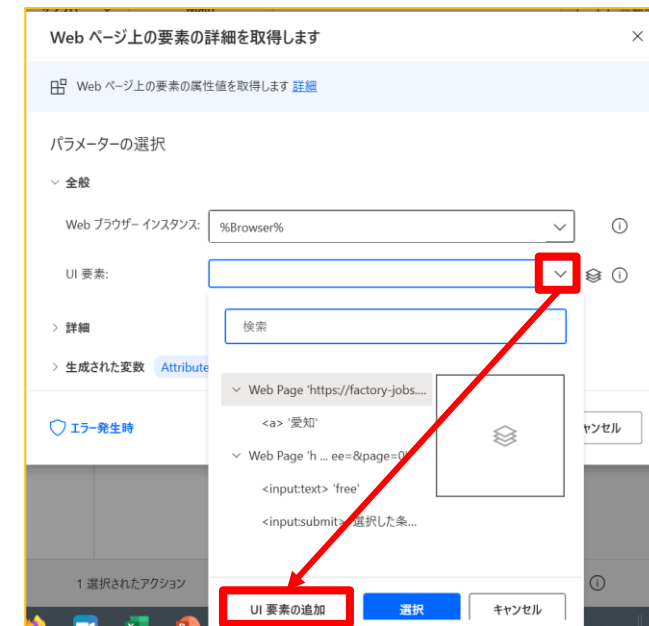
保存 キャンセル

① Webページのデータ取得

ブラウザ自動化→Webデータ抽出
→Webページ上の要素の詳細を～



UI要素の追加をクリック



① Webページのデータ取得

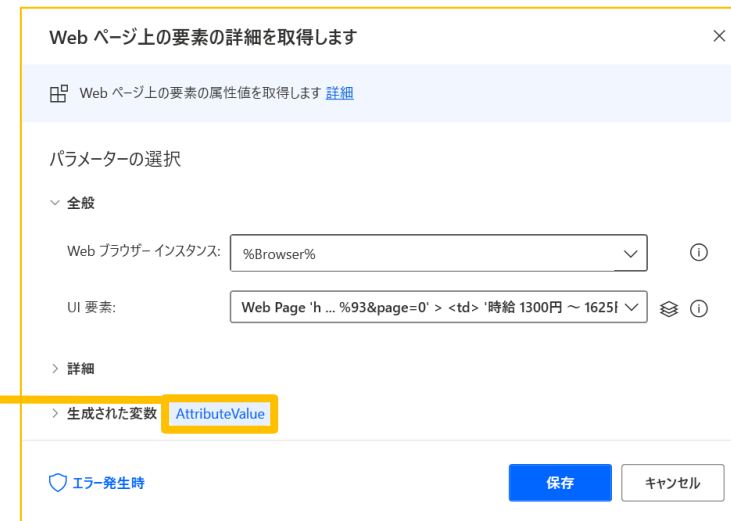
Webページ上の取得したいUI要素にポインターを合わせると赤い枠が表示されます
Ctrlを押しながらクリック



追跡セッションに
UI要素が追加されます

データが取得できて
いるか、実行して
確認してみましょう

取得したデータが
%AttributeValue%に
格納されます



① リストやテーブルのデータを一括取得

構造化された項目は2か所指定することで最後まで一括してデータを取得することができます

ブラウザ自動化→Webデータ抽出
→Webページからデータを抽出する

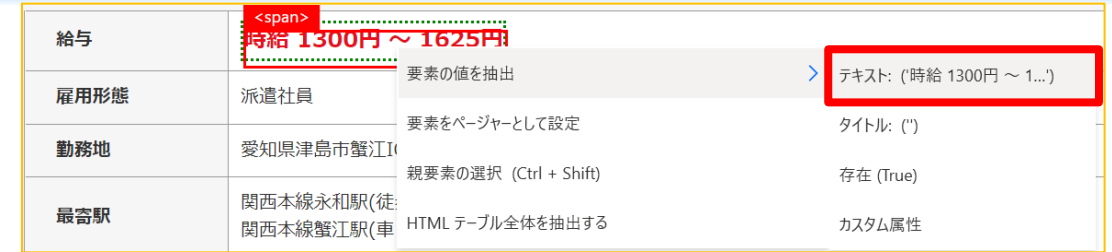
The screenshot shows the 'Web データ抽出' (Web Data Extraction) menu in the left sidebar, with 'Web ページからデータを抽出する' (Extract data from web page) highlighted in red. A red arrow points from this menu item to a detailed view of the action. In this view, a yellow box labeled '削除' (Delete) points to the first action in the list. Another yellow box highlights the second action, 'Web ページ上の要素の詳細を取得します' (Get details of elements on the web page), which is the action being configured in the dialog below.

変数 | 取得したデータが変数に格納されます
Excelスプレッドシート | 検索結果がExcelとして開きます

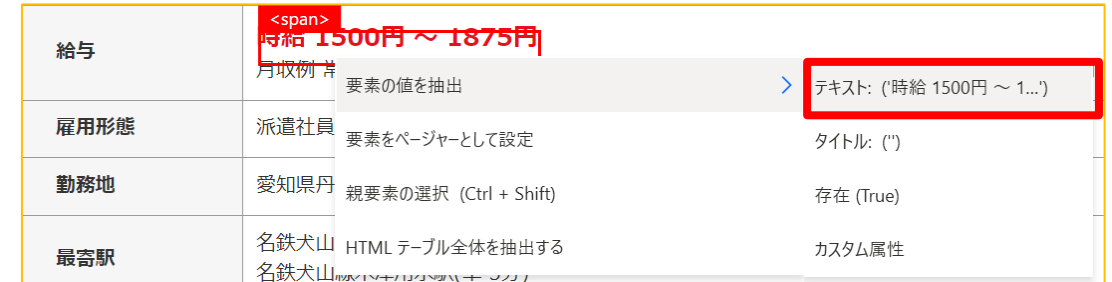
The dialog box 'Web ページからデータを抽出する' (Extract data from web page) is shown. It includes a description of the action, a 'パラメーターの選択' (Parameter selection) section with 'Web ブラウザー インスタンス' (Web browser instance) set to '%Browser%' and a 'データ保存モード' (Data save mode) dropdown menu. The dropdown menu is open, showing '変数' (Variable) and 'Excel スプレッドシート' (Excel spreadsheet) as options. The '変数' option is highlighted in yellow. At the bottom, there are buttons for 'エラー発生時' (When error occurs), '保存' (Save), and 'キャンセル' (Cancel).

① リストやテーブルのデータを一括取得

【1 件目】の取得したいUI要素を
右クリック→要素の値を抽出→テキスト



【2 件目】の取得したいUI要素を
右クリック→要素の値を抽出→テキスト



すると3件目以降もリストとして認識
してすべて取得することができます



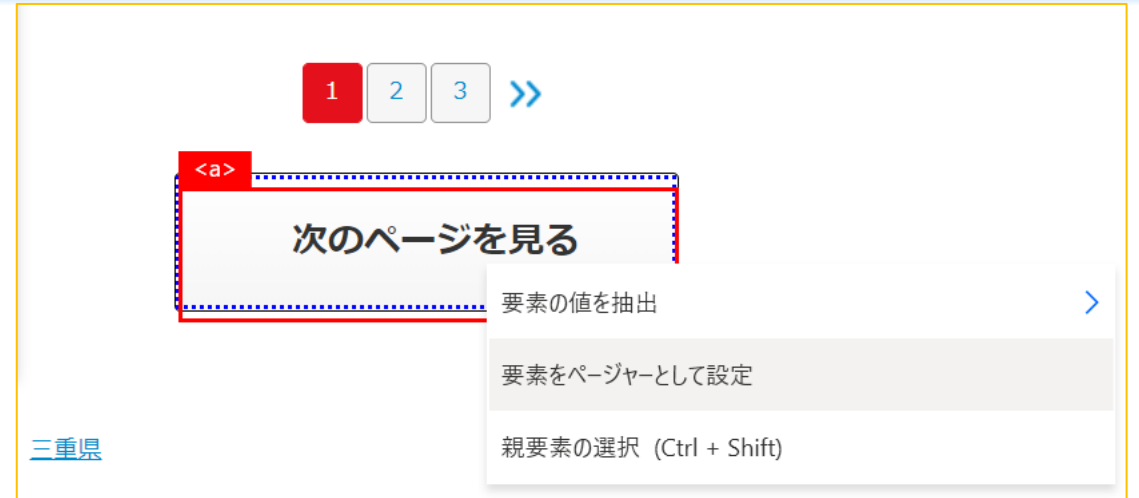
① リストやテーブルのデータを一括取得

引き続き、1件目に戻り、別の項目で
右クリック→要素の値を抽出→テキスト
を選択することで一気にデータを取得
することができます

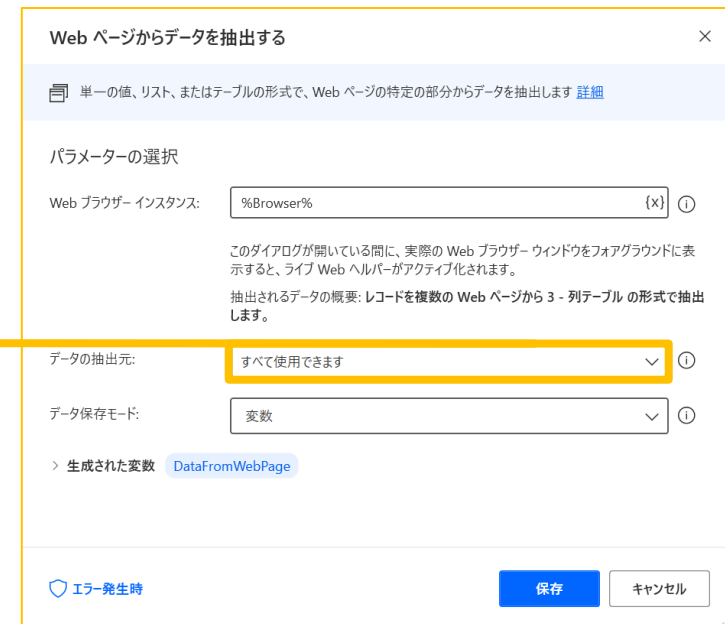


① ページをまたいでデータを一括取得

「次のページ」を右クリック→要素を
ページャーとして設定



最初のみ | 取得するページ数を指定できます
すべて使用できます | 最終ページまで取得を繰り返します



[参考]

① [参考]演算子

演算子	最初のオペランドと 2番目のオペランドの関係
=	同じ
<>	違う
>	最初が2よりも大
<	最初が2より小
>=	最初が2以上
<=	最初が2以下
次を含む	最初の中に2と同じ値が含まれている
次を含まない	最初の中に2と同じ値が含まれていない
先頭	2からはじまっている
先頭が次でない	2からはじまっていない
末尾	2でおわっている
末尾が次でない	2でおわっていない

① [参考]カスタム日時形式文字列

文字列	説明	使用例
yy	年 0～99	2024年→24
yyyy	年 0000～9999	2024年→2024
M	月 0～12	4月→4 12月→12
MM	月 00～12	4月→04 12月→12
d	日 0～31	1日→1 31日→31
dd	日 00～31	1日→01 31日→31
ddd	曜日の略称	金曜日→金
dddd	曜日の完全名	金曜日→金曜日

文字列	説明	使用例
h	12時間形式 時 0～12	1時→1 13時→1
hh	12時間形式 時 00～12	1時→01 13時→01
H	24時間形式 時 0～24	1時→1 13時→13
HH	24時間形式 時 00～24	1時→01 10時→10
m	分 0～59	5分→5
mm	分 00～59	5分→05

① [参考]値の取り出し方 (Datatable)

列名を[]の中に入れる時...
'(半角アポストロフィ)で囲む

変数を[]の中に入れる時...
%%や"は不要

指定したい列

		列名	左から×列目	変数 (%NewVar2%)
指定したい行	行数	% Datatable名 [行数]['列名'] % (例) %ExcelData[0]['列1']%	% Datatable名 [行数][列数] % (例) %ExcelData[0][0]%	% Datatable名 [行数][変数] % (例) %ExcelData[0][NewVar2]%
	変数 %NewVar1%	% Datatable名 [変数]['列名'] % (例) %ExcelData[NewVar1]['列1']%	% Datatable名 [変数][列数] % (例) %ExcelData[NewVar1][0]%	% Datatable名 [行の変数][列の変数] % (例) %ExcelData[NewVar1][NewVar2]%

	(0	1	2	3)	←×列目
#	列1	列2	列3	列4	←列名
0	りんご1	みかん1	ぶどう1	レモン1	
1	りんご2	みかん2	ぶどう2	レモン2	
2	りんご3	みかん3	ぶどう3	レモン3	

りんご1 = %ExcelData[0][0] %
= %ExcelData[0]['列1'] %

① [参考]値の取り出し方 (List)

		指定したい列
		列の指定不要
指定したい行	行数	% リスト名 [行数] % (例) %List[0]%
	変数 %NewVar1%	% リスト名 [変数] % (例) %List[NewVar1]%

※リストに列の概念はありません
(必ず一列のみである為)

(間違った例) %List[0]['アイテム']
%List[0][0]

	#	アイテム
行数 →	0	りんご1
	1	りんご2
	2	りんご3

りんご1 = %List[0]%

① [参考]値の取り出し方 (Datarow)

		指定したい列		
		列名	左から×列目	変数 (%NewVar2%)
指定したい行	行の指定 不要	% Datarow名 ['列名'] % (例) %CurrentItem['列1']%	% Datarow名 [列数] % (例) %CurrentItem[0]%	% Datarow名 [変数] % (例) %CurrentItem[NewVar2]%

※Datarowに行の概念はありません。
(必ず一行のみである為)

(間違った例) %CurrentItem[0]['列1']%

りんご 1 = %CurrentItem[0]%
= %CurrentItem['列1']%

(0	1	2	3)	←×列目
列1	列2	列3	列4	←列名
りんご1	みかん1	ぶどう1	レモン1	

① [参考] LoopとForEachの値の取り出し方の違い

アクション:Excelワークシートから読み取り

氏名	性別	生年月日
〇〇 〇〇	男性	1990/01/05
△△ △△	女性	1980/02/18
□□ □□	男性	1995/03/26



変数の値

ExcelData (Datatable)

#	Column1	Column2	Column3	Column4
0	0	氏名	性別	生年月日
1	1	〇〇 〇〇	男性	1990/01/05 0:00:00
2	2	△△ △△	女性	1980/02/18 0:00:00
3	3	□□ □□	男性	1995/03/26 0:00:00

① [参考] LoopとForEachの値の取り出し方の違い

Loop 指定した回数分の繰り返し処理

Loop

指定された数のアクションのブロックを反復処理します [詳細](#)

パラメーターの選択

開始値: 1 (X) (i)

終了: 3 (X) (i)

増分: 1 (X) (i)

> 生成された変数 LoopIndex

保存 キャンセル

生成される変数 %LoopIndex%(数値型)

1回目の処理= 『1』

2回目の処理= 『2』

#行番号が連番なのでLoopアクションに
置き換える事ができる

ForEach 指定した変数の要素の 数だけ反復処理

For each

リスト、データテーブル、またはデータ行にあるアイテムを反復処理して、アクションのブロックを繰り返して実行します [詳細](#)

パラメーターの選択

反復処理を行う値: %ExcelData% (X) (i)

保存先: CurrentItem (X)

保存 キャンセル

生成される変数 %CurrentItem%(DataRow型)

1回目の処理= 『0 氏名 性別 生年月日』

2回目の処理= 『1 〇〇〇〇 男性 1990/01/05
0:00:00』

CurrentItemの中には
[#][Column1]~[Column4]が入る