Microsoft Power Automate for desktop 活用講座



※Microsoft アカウントの下記の情報が必要ですので、準備の上ご参加ください。

- ✓ メールアドレス
- ✓ パスワード

Windows パソコンをお使いであれば、購入時のセットアップの際に Microsoft アカウントを作成 していると思います。パソコンの説明書など購入時の書類一式を改めてご確認ください。もしわか らない場合でも、本講座で新規作成方法をレクチャーいたしますので、安心してご参加ください。

目次

1	動作環境	4
2	Power Automate for desktop とは	4
3	クラウド版について	5
4	Microsoft アカウントの作成	6
5	Power Automate for desktop をインストールする11	2
6	Power Automate for desktop の起動と終了10	6
	6.1 Power Automate for desktop の起動方法	6
	6.2 Power Automate for desktopの終了方法19	9
7	Power Automate for desktopの画面構成について20	0
	7.1 コンソール20	0
	7.2 フローデザイナー	3
8	新しいフローの作成と実行2	5
	8.1 メッセージボックスの作成と実行2	5
	8.2 フローを実行する 2 つの方法2	8
9	変数について3	1
	9.1 変数とは	1
	9.2 変数を作る 2 種類の方法	2
	9.3 変数のデータ型について	3
	9.3.1 数值型	4
	9.3.2 テキスト型	6
	9.3.3 DateTime 型	7
	9.3.4 ブール値型	0
	9.3.5 リスト型5	1
	9.3.6 データテーブル型	0
1() 繰り返し処理	0

	10.1 Loop アクション	71
	10.2 For each アクション	85
11	条件分岐	98
	11.1 if アクション	98
	11.2 Else アクション	. 102
	11.3 Else if アクション	. 104
	11.4 「if」「Else if」「Else」の動作イメージ	. 110
	11.5 オペランドと演算子	. 111
	11.6 「メッセージを表示」アクションで変数に格納される値について	. 112
12	作成したフローを共有する方法	. 114
13	最後に	. 116

1 動作環境

本講座では 2023 年 1 月時点での下記の動作環境で問題なく動作することを確認しております。

- Windows 10 Home バージョン 22H2
- Power Automate for desktop バージョン 2.27.177.22340
- Microsoft Excel 2016

上記に加えインターネットに接続した環境と Microsoft アカウント(メールアドレスとパスワード) が必要です。Windows パソコンをお使いであれば、購入時のセットアップの際にアカウントを作成して いる思います。パソコンの説明書など購入時の書類一式を改めてご確認ください。もしわからない場合 は「Microsoft アカウントを作成」のセクションにて新規作成方法をレクチャーいたします。

尚、本講座における操作説明の手順や各種画像は、2023 年 1 月現在の内容を反映したものです。IT 業 界は変化のスピードが速く、Microsoft 社により操作手順や画面レイアウトなどが変更される可能性が あるため、常に最新の状況を反映した内容ではないことをご容赦ください。

2 Power Automate for desktop とは

Power Automate for desktopとは、Microsoft社が提供する業務を自動化する RPA ツールです。 RPA とは、Robotic Process Automation の頭文字を取った言葉で、定型的な事務作業をソフトウェア によって自動化することです。RPA は「ノーコード」や「ローコード」などと呼ばれることもあり、プ ログラミングを全く行わない(ノーコード)、あるいは少しだけ行う(ローコード)だけで業務を自動化 できることから付けられた名称です。

Power Automate for desktop の利点は3つあります。

- 1. プログラミングスキルがない人でも扱えるくらいに操作が簡単。
- Windows 11 にはデフォルトでインストールされていて、Windows 10 でも無料でインストールして利用できる。
- 3. Microsoft 公式のアプリケーションであるため Excel や Word などとの連携がしやすい。

特に2番目の利点である無料で使える点は重要です。他社の RPA ツールには、数十万円~数百万円と 高価なものが多い中、無料で使えるのは大きな利点です。とは言え、実は Power Automate には有料版 も用意されていて、クラウド版などと呼んだりします。次のセクションにて説明いたします。

3 クラウド版について

Power Automate for desktop は Microsoft 社が提供している「Power Automate」の一種です。 Power Automate には「デスクトップフロー」と「クラウドフロー」の2種類があり、デスクトップフ ローを担当しているのが Power Automate for desktop というわけです。デスクトップフローは、自 身のパソコンに Power Automate for desktop をインストールし、クラウド上ではなく、デスクトッ プ上で操作を行います。

そして、もうひとつがクラウドフローです。単に「Power Automate」と呼んだ場合は、こちらのク ラウドフローのことを指すことが多いです。クラウドフローは一部無料で使える機能もありますが、基 本的には有料で使うことを想定しています。有料版のクラウドフローでは「コネクタ」と呼ばれる他のク ラウドサービス (Twitter や Salesforce など)と連携する機能を利用してクラウド上で業務の自動化を 行うことができたり、スケジュール実行やトリガー実行などの機能が使えたりします。

本講座では、無料の Power Automate for desktop を使ったデスクトップフローを学習していきます。

デスクトップフロー	クラウドフロー						
Power Automate for desktop	Power Automate						
自身のパソコン上で Excel、Word、Web ブラウザ、ファイルなどの操作を自動化	クラウド上で Microsoft365 や Twitter、 Salesforce などの様々なクラウドサー						
できる。	ビスと連携できる。						

4 Microsoft アカウントの作成

すでに Microsoft アカウント (メールアドレスとパスワード) をお持ちの場合は、このセクションは 飛ばして「5. Power Automate for desktop をインストールする」へお進みください。また、現在のご 自身の Microsoft アカウント情報をお忘れの場合は、下記の方法でご確認ください。特にパスワードの リセットには多少複雑なプロセスが必要です。尚、Microsoft アカウントは同じ人が複数作成することも 可能ですので、現在所有のアカウント情報がどうしても思い出せない方は、新規に作成した方が早いか もしれません。

- Microsoft アカウント(メールアドレスとパスワード)の確認方法 https://nekosaya.com/check-account/
- Microsoft アカウントのパスワードのリセット方法 https://bit.ly/3uPR0m7

それでは、ここから Microsoft アカウントを新規に作成していきましょう。

① Microsoft Edge ブラウザを起動する



② Microsoft アカウントの Web ページにアクセスする



③ 「アカウントを作成」をクリック



④ 新しいメールアドレスを作成する





⑤ パスワードを作成する

Microsoft				
	●ご自身で考えた新しいパスワード を入力。			
パスワードの作成	パスワードは8文字以上で、大文字、 小文字、数字、記号のうち2種類以上 を含んでいる必要があります。			
お客様のアカウントで使用するパスワードを入力します。	また、今後何度も必要になりますの で、忘れないようどこかに控えておき ましょう。			
パスワードの表示				
Microsoftの製品とサービスに関する情報、ヒント、お よびキャンペーンのメール受信を希望します。				
[次へ] を選択することにより、Microsoft サービス規約とプライバシーと Cookie に関する声明に同意するものとします。 次へ	❷「次へ」をクリック			

⑥ ロボットでないことを証明する





Microsoft	
サインインの状態を維持しますか?	③クイズに正解すると「サインイン状態を維持しますか?」と聞かれるので「はい」をクリック
必要がなくなります。 ○ 今後このメッセージを表示しない	
いいえ はい	

⑦ Microsoft アカウントの管理画面でユーザーの名前を設定する

iii Microsoft アカウント あなたの情報 プライバシー	セキュリティ Rewards	支払いと課金情報 ~	サービスとサブスクリプション	ज्ञारनत्र ? s
名前を追加する	❶Microsoft ア 「名前を追加す	Microsoft 365の入手 サービスとサブスクリブ カウントの管理 る」をクリック	^{ション} <i>2 パスワード</i> セキュリテー 目画面が表示されます	を変更する イ ので
名前を編集する	×			
姓		2 「姓	٤」の欄に名字を入力	
名		③ 「名	」の欄に名前を入力	
Captcha				
VA		❹表示 く読み	、されているグニャグ →取って、ここに入力。	ニャの文字を正し 。
() 新規 Φ)) オーティオ 表示された文字を入力		左画像	えては「SPVR」と入れ	りできれば正解。
保存	キャンセル	⑤ 「伢	そう」をクリック	



これで Microsoft アカウントの新規作成は完了です。今回設定したメールアドレスとパスワードは今後学習を進めていく上で何度も必要になりますので、人の目の触れないところで大切に保管しつつ、忘れた際にすぐに取り出せるようにしておきましょう。

5 Power Automate for desktop をインストールする

Windows 11 には Power Automate for desktop が標準でインストールされています。次のセクション「6. Power Automate for desktop の起動と終了」へお進みください。このセクションでは Windows 10 のユーザー向けに Power Automate for desktop をインストールする方法について解説いたします。

① 公式インストールページへアクセスし、インストーラーをダウンロード後、インストールする

①Microsoft Edge ブラウザを起動し、アドレスバーに

<u>https://learn.microsoft.com/ja-jp/power-automate/desktop-flows/install</u>

と入力し Enter を押す。あるいは「Power automate インストール」と検索し、上記と同じ URL のページにアクセ

スしてください。すると下図の公式ページが開かれます。









② Microsoft Edge に Power Automate の拡張機能をインストールする



拡張機能はインストールしただけでは機能しないため、有効化する必要があります。そのため、有効化されているか どうかを確認します。



これで Power Automate for desktop のインストールが無事成功しました。次のセクションでは Power Automate for desktop の起動と終了の方法について学習していきます。

6 Power Automate for desktop の起動と終了

このセクションからは Windows 10 と 11 共通の内容となっておりますが、講義画像などは Windows 10 のものを使っておりますので Windows 11 ユーザーの方は適宜読み替えて操作してください。

6.1 Power Automate for desktop の起動方法

それでは Power Automate for desktop の起動方法から学習していきましょう。前セクションにて「ショートカットを作成」にチェックを付けた方はデスクトップにあるショートカットをダブルクリックするだけで起動できます。その場合は「サインイン画面が開かれました」のステップ④から開始してください。ここではショートカットがない方や Windows 11 の方のために、より一般的な起動方法を紹介いたします。

① Power Automate for desktop のサインイン画面を立ち上げる



※この画像は Windows 10 のもの	ですので11の方は適宜	主読み替えてください
すべて アプリ ドキュメント ウェブ その	D他 ▼	в … ×
最も一致する検索結果		
Power Automate アプリ		
Web の検索	_ \	Power Automate
🔎 power automate - Web 結果を見る	>	アプリ
	>	
		愛検案欄に 'power duroniare」 2 入力
Power Automate for desktop_sample	>	③検索結果の中の「Power Automate」のアイ
tmp_power_automate	>	コンをクリック







これで Power Automate for desktop の起動が完了しました。尚、2回目以降の起動ではメールアド レスとパスワードの入力だけ行えばサインインできます。次に終了方法について学習します。今の画面 を開いたままの状態にしておいてください。



これで Power Automate for desktop の起動方法と終了方法の学習が完了しました。今後学習を進める上で何度も行う操作ですので、よく覚えておきましょう。

7 Power Automate for desktop の画面構成について

前回までのセクションにより Power Automate for desktop の利用環境が整いました。次は、Power Automate for desktop の画面構成について学習しましょう。結論から言うと、Power Automate for desktop はフローの実行と管理を行う「コンソール」と、フローの作成と編集を行う「フローデザイナー」という2つの画面から構成されています。

具体的な説明の前に、この後頻繁に出てくる下記2つの用語を覚えておきましょう。

■ アクション

Power Automate for desktop が標準で用意してくれている、パソコン上でよく行う操作のことで、 300 種類以上用意されています。例えば「Excel の起動」や「メールの送信」などがあります。

■ フロー

複数のアクションを並べたものです。自動化したい業務の流れに沿ってアクションを組み合わせて いき、でき上がった1つのまとまりのことです。

それではまず「コンソール」から具体的に見ていきましょう。

7.1 コンソール

まず、前セクションにて学習した方法で Power Automate for desktop にサインインしておきましょう。すると下図のようなホーム画面が表示されているはずです。表示されていない方は、画面左上の「ホ ーム」をクリックしてホーム画面へ移動しておいてください。それでは、コンソール画面へと移動してみ ましょう。



これがコンソール画面です

	ower Automate	pclearning smo 👘	□ ×
+ 新しいフロー	资源 资源 资金 资金	? ヘルプ 🔎 フローの検索	
ホ-ム 自分のフロ- 例	€ 900. **	() 量新の状態。	「更新
·. 	(†) (†)		
70-	なし		
フローが何もありません。最初	こフローを作成してください。		
+ #iL	-07		

現在はフローをひとつも作成していないので「フローなし」と表示されていますが、フローを作成する とコンソールは下図のような画面になります。

●新しいフロー	0	サインインアカウント名
+ 新いノフロ- ▶ 実行 □ 停止 🖉 編集 …	Power Automate 3設;	EBERSE - □ × EBERSE - □ × E E E E E C S E C C C C C
ホーム 自分のフロー 例	❹フローの一覧	○ 最新の状態に更新
○ _o / ^a 名前	変更日時 ↑	ステータス
● ●	D 2 : 6日前	実行されていません
₩ebサイトログインリトライ	6 日前	実行されていません

❶新しいフロー

新しいフローを作成します。

❷サインインアカウント名

Power Automate for desktop に現在サインインしているアカウント名が表示されます。

₿設定

Power Automate for desktop の各種設定を行います。

❹フローの一覧

これまでに作成したすべてのフローの一覧が表示されるエリアです。

「フローの一覧」をもう少し詳しく見てみましょう。



€実行

フローを実行します。

❷停止

実行中のフローを途中で停止させます。

❸編集

フローを編集するための「フローデザイナー」を起動させます。

❹その他のアクション

フローの削除、フローの名前の変更、フローのコピーができます。

6フローの状態

現在のフローの状態が表示されます。上図の他に「実行中」などがあります。

現在の「フローなし」と表示されたコンソールを開いたままにしておいてかまいません。この 2 つ後の「8.新しいフローを作成しよう」では、この画面からスタートします。

7.2 フローデザイナー

フローデザイナーは、アクションを組み合わせてフローを作成していくための画面です。コンソールに て「新しいフロー」あるいは「編集」をクリックした際に表示されます。フロー作成に必要な機能が一画 面にコンパクトにまとめられており、直感的に操作できる画面構成になっています。Power Automate for desktop を使う上で特に重要なのは、画面左側の「アクションペイン」、中央の「ワークスペース」、 右側の「変数ペイン」です。

	メニューバー	④ ツール	バー ⑤ Ma	in タブ	⑦変	数ペイン	❸ UI 要素ペイ	ン
	ファイル 編集 デバッグ ツール	表示 ヘルプ		sample Power Automate			- 0	×
די ג	クション アクションの検索	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	l 		م	変数	×	x}
	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	⑥ サブフロータ	ブ	÷		入出力変数 0 ここにはまだ入力さ	または出力の変数がありません ①	
> 5 > 7 > 7 > 7	ワークステーション スクリプト ファイル フォルダー					∨ 7□-変数 0	る変数がありません	
> E > L > F > 5	王縮 JI オートメーション HTTP ブラウザー自動化		⑦ワークス ここにはまだアクション	ペース ンがありません				
> E > 7 > 2	Excel データベース メール	E	開始するには、このペインにアクシ: プするか、レコーダーを使用して;	ョンをドラッグ アンド ドロッ アクションを取得します				
> C > J > J	xxrange server Dutlook メッセージ ボックス マウスとキーボード							
\odot	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 🗘 訂秒 🛈)		
		_						
0 7	クションペイン			⑨ 状態バー				

①メニューバー

フローの実行や保存など、フローの作成に必要な各種操作のボタンが並んでいます。クリックすると詳 細コマンドが展開されます。また「ヘルプ」ボタンから開ける公式ドキュメントを読むことで、さらに 専門的な内容を調べることができます。

②アクションペイン

自動化に必要な部品である「アクション」がグループごとに分類され並んでいます。先頭の「>」印を クリックすると詳細なアクションが展開されます。

❸ワークスペース

ドラッグアンドドロップにより、ここにアクションを並べてフローを作成していきます。

④ツールバー

フローの保存、実行、停止、デバッグモードでの実行、レコーダー機能のボタンが並んでいます。

23 / 116

⑤Main タブ

サブフローと違い、Main タブに配置したフローは実行したときに必ず最初に実行されます。削除や名前の変更はできません。

〇サブフロータブ

サブフローの一覧を表示します。サブフローの用途として例えば、Main タブにて頻繁に実行するフロ ーがあった場合、同じ記述を何度も書くのは面倒なので、1 回の実行分だけサブフローに記述してお き、それを Main タブから何度も呼び出すといった使い方をします。Main タブから何度も呼び出すと は言え、先ほどと違い今度はサブフロー名を記述するだけで呼び出せる分、記述量が少なくて済むので す。

⑦変数ペイン

フローで使用するすべての変数が表示されます。変数の検索や変数に格納された値の確認などができ ます。変数については後ほど学習します。

❸UI 要素ペイン

フローで使用する UI 要素の管理を行います。UI とは User Interface のことで、お問い合わせフォームなどでよく見かける、文字入力をするボックスや、チェックボックス、ドロップダウンリスト、送信ボタンなど、ユーザーが直接操作する部分のことです。

9状態バー

フローのステータス、選択されたアクション、フロー内のアクション、サブフローの合計数などが表示 されます。フロー実行中には、実行開始からの経過時間が、エラーがある場合はエラー数が表示されま す。フローの動作テストを行いたいときに重宝します。

8 新しいフローの作成と実行

ここでは、練習用に簡単なフローを作成していきます。現在、「フローなし」と表示されたコンソール が開かれた状態だと思います。そうでない方は「7.1 コンソール」のセクションを参考に、コンソールを 開いた状態にしておいてください。

8.1 メッセージボックスの作成と実行

① 新しいフローの作成







これで「mysample」という名前の新しいフローを作成することができました。しかし、中央のワーク スペースの箇所に「ここにはまだアクションがありません」と書いてある通り、まだ「フローの入れ物」 ができた状態に過ぎません。アクションを配置してフローを完成させてみましょう。ここでは「メッセー ジボックス」のアクションを配置します。



すると「メッセージを表示」のダイアログボックスが自動的に開かれるので、下記のようにアクション の内容を設定していきます。

メッセージを表示			×	
💭 メッセージ ボックスを表示します 詳細				
パラメーターの選択				・メッセージボックスのタイトル」
~ 全般				欄に「テストメッセージ」と入力
メッセージ ボックスのタイトル:	テストメッセージ	{x} (j)	
表示するメッセージ:	これはテストです。	{x} (j)	
				④「表示するメッセージ」欄に「これ はテストです。」と入力
メッセージ ボックス アイコン:	いいえ	~ C		
メッセージ ボックス ボタン:	ОК	~ (i)	
既定のボタン:	最初のボタン	~ C)	❺「メッセージボックスを常に手前に
メッセージ ボックスを常に手前に表示する:		©	,	表示する」をオンにします。この画像 と同じ状態がオンです。
メッセージ ボックスを自動的に閉じる:		C)	
> 生成された変数 ButtonPressed				❻「保存」をクリック
○ エラー発生時	[保存 キャンセル		

⑦「メッセージを表示」アクションがワークスペース に追加されました

ファイル 編集 デバッグ ツ	ール 表示 ヘルプ			mysample Power Automate
アクション		۲		م
🔎 アクションの検索	√ שדעד יי	Main		
> 変数 > 条件 > ループ	1	メッセージを表示 夕イトルが 'テストメッ・ たボタンを Button	セージ' である通知; Pressed に保存(ポップアップ ウィンドウにメッセージ 'これはテストです。 ' を表示し、押され : よす
	❸ツールバーの	の「保存」をクリッ	クしてフロ	ーを保存

これで新しいフローの作成は完了しました。次に、実際にこのフローを実行してみましょう。

② フローの実行

ファイル 編集 デパッグ	ツール 表示 ヘルフ	7	mysample Power Automate
アクション			●ツールバーの「実行」をクリック
🔎 アクションの検索	₀⁄º サブフロ- 〜	Main	
> 変数> 条件	1	メッセージを表示 ワイトルが 'テストメッt たボタンを ButtonP	2ージ' である通知ポップアップ ウィンドウにメッセージ 'これはテストです。' を表示し、押され : ressed に保存します
> ループ			



これで、無事フローの実行まで完了しました。

8.2 フローを実行する 2 つの方法

フローを実行する方法には「デバッグモードで実行」と「本番モードで実行」の2種類があります。こ こで2つの実行方法について整理しておきましょう。

① デバッグモードで実行

フローデザイナー上で実行することを「デバッグモードで実行」といいます。前回のメッセージボッ クスを表示する事例がまさにそうです。

ファイル 編集 デバッグ ツ	ノール 表示 ヘルプ		mysample Power Automate
アクション		◉	フローデザイナー上の「実行」をクリック
🔎 アクションの検索	₀⁄º サブフロ- 〜	Main	▶デバッグモードで実行
> 変数 > 条件	1	メッセージを表示 ↓ タイトルが 「テストメッセージ」 たボタンを ButtonPressed	である通知ボッブアップ ウィンドウにメッセージ 'これはテストです 。' を表示し、押され : d に保存します
> ループ			

デバッグとは、プログラムの誤りを特定し修正する作業のことです。フロー作成中にデバッグするこ とで、本番運用の際にエラーが発生するリスクを減らすことができます。 ② 本番モードで実行

コンソール上で実行することを「本番モードで実行」といいます。フローデザイナーを起動する必要 がない上に、動作速度も速いという利点があります。前回のメッセージボックスを表示するフローを 本番モードで実行してみましょう(フロー名は mysample でしたね)。

 ●現在いるフローデザイナーの ルの画面へ移動します。 ※画面下部のアイコンにマウス 像のように小さな画面が表示 	の画面からコンソー を乗せると、この画 されますのでコンソ	
ールの画面をクリック。	Power Automate	mysample Power Automate
o 選択された	A R 20 X X	

	Power Automate	pclearning smo 🚽 🗆 🗙
┿ 新しいフロー ▷ 実行	- □ 停止 🖉 編集 …	🛞 設定 ? ヘルプ 🔎 フローの検索
ホーム 自分のフロー 例 	❷コンソール上の「実行」をクリック ▶本番モードで実行	▶ 最新の状態に更新
✓ ₀/⁰ 名前	変更日時 ↑	ステータス
🕑 🖵 mysample		現在編集中

	Power Automate	pclearning smo 🚽 🗆 🗙
┿ 新しいフロー Ⅱ 一時停止 🗌 停止		設定 ? ヘルプ 🔎 フローの検索
ホーム 自分のフロー 例 		💍 最新の状態に更新
🥑 ⊳⁄° 名前	変更日時 ↑	ステータス
🕑 🖵 mysample	🗌 🧷 : 21 時間前	実行中
	テストメッセージ × これはテストで す。 	ジボックスが表示されます 5 クリック

このとき、下記のように画面右下に「Windows の通知」が表示されます。デバッグモードほどでは ないですが、実行状況を簡単に知らせてくれているわけです。

Power Automate	×	Power Automate ×
フロー 'mysample' が実行中です		フロー 'mysample' が正常に完了しました
一時停止	停止	実行時間: 00:00:59

③ デバッグモードと本番モードの使い分け

本番モードは、実行するひとつひとつのステップを確認できないため、エラーが発生した際の原因を 特定しづらいという欠点があります。そのため、下記2つに留意してフローを作成しましょう。 (1)フローデザイナーのデバッグモードで実行しエラーがないことを確認しておく

(2)エラーが発生したときの対処をしておく

以上、新しいフローを作成し 2 つの方法で実行してみました。フローデザイナーの画面右上の¥印を クリックしてフローデザイナーを閉じましょう。すると、コンソール画面が表示されていると思います。 再びフローデザイナーを開き、フローを編集したい場合は編集ボタンをクリックしてください。

	Power Automate	pclearning smo 😑 🗆 🗙
┿ 新しいフロー ▷ 実行	□ 停止 🧪 編集 …	😳 設定 ? ヘルプ 🔎 フローの検索
		ここをクリックしてサインアウト
ホーム 自分のフロー 例 	再びフローデザイナーを開きたいときは 「編集」をクリック	最新の状態に更新
✓ ₀/º 名前	変更日時 ↑	ステータス
🕑 🖵 mysample	□ □	実行されていません

最後に「Power Automate for desktop の起動と終了」のセクションで学習した方法でサインアウト しておきましょう。これでこのセクションは終了です。

9 変数について

Power Automate for desktop を使いこなすには「変数」の理解が重要です。少し難しく感じるかも しれませんが、自動化処理を活用するためには必要な内容です。このセクションでは、変数の基本的な使 い方について学んでいきます。

9.1 変数とは

変数とは「データ(値)を一時的に保存しておくための名前付きの箱」のことです。変数の名前のこと を「変数名」といい、Power Automate for desktop ではパーセント記号(%)で前後を囲む必要があ ります。下図では、「%name%」という名前の箱に「鈴木花子」という値を保存(格納)し、「%age%」 という名前の箱に「65」という値を保存(格納)しています。このように、データを箱の中に保存するこ とで何度でも取り出し再利用が可能になるわけです。



それから、ここでは変数に値を入れることを「保存」や「格納」と表現していますが、教科書によって は「変数に値を代入する」「変数に値をセットする」「変数に値を設定する」などと表現される場合もあ ります。混乱するかもしれませんが、意味はすべて同じということを覚えておきましょう。

また、変数名として使用できるのは半角英数字とアンダースコア(_)だけです。尚、数字は使用可能 ですが、数字から始まる変数名は設定できません。つまり例えば「%test_1%」は可能ですが「%1_test%」 は不可ということです。

9.2 変数を作る 2 種類の方法

Power Automate for desktop では、変数は2種類の方法で作ることができます。自分で作る方法と 自動で作る方法です。ここからの操作では、前回作成したフロー「mysample」をそのまま使っても構い ませんし、新しいフローを作成してもどちらでも構いません。「mysample」を使う場合は前回作成した 「メッセージボックスを表示」アクションは削除し、ワークスペースに何もない状態にしておきましょ う。

① 「変数の設定」アクションを使って自分で作る方法

フローデザイナーを開き、ワークスペースに変数グループの「変数の設定」 アクションをドラッグ あるいはダブルクリックにより配置します。



<u>変数の設定</u> ×	❷「変数の設定」ダイアログボッ クスが開かれました。
{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	
変数: NewVar {x} 值: {x} ①	 ⑦「変数: NewVar」と表示され ているところは変数名、つまり 箱の名前を設定するところで
	9 o
	④「値:」と表示されているところは変数の値、つまり箱の中身を設定するところです。
保存 キャンセル	

変数の設定	×	●変数名 NewVar は変更可能 です。NewVar の箇所をクリッ のよう「Tan Data 」と、たまえの
{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 <u>詳細</u>		クレーIaxKate」と入力して変 更してみましょう。
変数: 《TaxRate%》 ×}		●値は「0.08」と入力
	×1 (1)	
		❼「保存」をクリック
保存 キャン・	214	

ファイル 編集 デパッグ ソ	ソール 表示 ヘルプ		mysample Power Automate			- 0	×
アクション		۱		م	変数	×	{x}
🔎 アクションの検索		Main			🔎 変数の検索		0
✓ 変数 > データテーブル	1	{x} 変数の設定 _{変数} TaxRate	に値 0.08 を割り当てる	÷	◇ 入出力変数 0		3
 (1) 数値の切り捨て (2) 乱数の生成 (2) 見、切ストのクリア (3) リストから項目を削除 (3) 取りたがが増き 			3 ワークスペースに「変数の言 定」アクションが追加されま	設 し	ここにはまだ入力または出力の変数があり	Jません	
1 ^{- 一} 見の並べ者だ 9三 リストのジャッフル ① リストの統合 ○ リストを逆にする ○ リストから重複する		; -	た。 変 数 ペ イ ン に も 変 犭 FaxRate が表示されます。	数	ソフロー変数 1 (1) TaxRate	V	

これで変数の設定が完了しました。つまり TaxRate という名前付きの箱を用意し、箱の中身として 0.08 という値を格納しました。しかし、作っただけでは変数の「威力」がよくわかりません。変数は使うことでその威力を発揮します。ここでは再びメッセージボックスを表示するアクションを設定して変数を使ってみましょう。

ファイル 編集 デバッグ ツ	ール 表示 ヘルプ	mysample Power Automate	
アクション			م
🔎 アクションの検索	_■ ⁄ª サブフロ- 〜	Main	
> Exchange Server	1	{x} 変数の設定 変数 TaxRate に値 0.08 を割り当てる	:
 > Outlook > メッセージ ボックス 		レ メッセージを表示	
メッセージを表示 入力ダイアログを表示		▲ スペークを表示」アクションを「変数の設定」アクションの下にドラッグして配置	

メッセージを表示			×	
💭 メッセージ ボックスを表示します <u>詳細</u>				
パラメーターの選択				
~ 全般				
メッセージ ボックスのタイトル:			{x} i	
表示するメッセージ:			(x) 🕕	②開かれたダイアログボックス にて「表示するメッセージ」の{x} マーク(変数の選択)をクリック
メッセージ ボックス アイコン:	変数の検索		í	
メッセージ ボックス ボタン:	名前	型	(j)	
既定のボタン:	✓ 70-変数 1 TaxRate	数值	Ū	 ③現れたドロップダウンリスト から、フロー変数の「TaxRate」
メッセージ ボックスを常に手前に表示する:			0	をクリックし「選択」をクリック
メッセージ ボックスを自動的に閉じる:	選択	キャンセ	<i>i</i>	
> 生成された変数 ButtonPressed	L			
○ Iラ-発生時		保存	キャンセル	
メッセージを表示			×	
パラメーターの選択				
~ 全般				④変数「%TaxRate%」が設定さ れたことを確認
メッセージ ボックスのタイトル: 表示するメッセージ:	%TaxRate%		{x} ()	これで変数を「使う」設定ができました
メッセージ ボックス アイコン:	いいえ		~ i	
メッセージ ボックス ボタン:	ОК		~ Ū	
既定のボタン:	最初のボタン		~ i	❺「メッセージボックスを常に手
メッセージ ボックスを常に手前に表示する:			Ū	前に表示する」を図のようにオン にします
メッセージ ボックスを自動的に閉じる:			(i)	J
> 生成された変数 ButtonPressed				❻「保存」をクリック
○ エラー発生時		保存	キャンセル	

ファイル 編集 デバッグ	ツ−ル 表示 ヘルプ		mysample Power Automate	
アクション		۲		
🔎 アクションの検索	_┏ ∕ª サブフロ- ∨	Main	⑦「変数の設定」アクションの下に 「メッセージを表示」アクションが	
> 変数	1	{x} 変数の設定 {x} 変数 TayBate に値 0.08 を割り当てる	追加されました	
> 条件				
> ループ	2			
> フロー コントロール	2	タイトルか である通知パップアック・ワイント・フレスツセーシ TaxKate を表示し、押されたパックを ButtonPressed に保存します		
> フローを実行する				

これで、変数を設定し、その変数に格納された値をメッセージボックスに表示するというフローが 完成しました。それではフローを実行してみましょう。

ファイル 編集 デバッグ	ツール 表示 ヘルプ	mysample Power Automate	
アクション			
🔎 アクションの検索	₀⁄º サブフロ- 〜	Main U「美行」をクリック	
> 変数	1	{x} 変数 TaxRate に値 0.08 を割り当てる	
> 条件 > ループ > フロー コントロール	2	メッセージを表示 ↓ タイトルが である通知ポップアップ ウインドウにメッセージ TaxRate を表示し、押されたボタンを ButtonPressed に保存します	
> フローを実行する			

ファイル 編	集 デバッグ ツ−ル 表示 ヘルプ	mysample Power Automate
日 保存 -	時停止 🗌 停止 🖂 次のアクションを	• VI-Ø-
_■ /ª サブフロー 〜	Main	
1	{x} 変数の設定 変数 TaxRate に値 0.08 を割り	り当てる
	メッセージを表示 タイトルがである通知ポップアップ	ウィンドウにメッセージ TaxRate を表示し、押されたボタンを ButtonPressed に保存します
	0.08	× ② メッセージボックスが表示されました
		●変数 TaxRate の値である 0.08 が 表示されていることが確認できます
		④「OK」をクリック

変数 ×	{×}	
🔎 変数の検索	8	
✓ 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 (+)		●画面右側の変数ペインにおいても TaxRateに0.08が格納されたことを確認 できます
√ 7□-変数 2		
(x) ButtonPressed OK		

◆変数の値は何度でも使用できる

一度変数に格納した値は何度でも取り出すことができます。それを確認するために「メッセージを 表示」アクションをさらにもう一つ追加して実行してみましょう。設定内容は先ほどの「メッセージ を表示」アクションとまったく同じですので参考にしてください。


6	2回目	が実	行されています		
	ファイル	編集	デバッグ ツール 表示	ヘルプ mysample Power Auto	mate
E 1	保存 ┃	一時	停止 🗌 停止 ▷ 次の	アクションを () レコーダー	م
۰⁄°	サブフロー	\sim	Main $ imes$		
	1		{x} 変数の設定 変数 TaxRate に値	0.08 を割り当てる	
	2			ポップアップ ウィンドウにメッセージ TaxRate を表示し、押されたボタンを ButtonPressed に保	存します
	3		メッセージを表示 タイトルがである通知	ポップアップ ウィンドウにメッセージ TaxRate を表示し、押されたボタンを ButtonPressed に保	存します
			●2回目のメッ	セージボックスが表示されました 0.08 のK	● 「OK」を クリック

◆変数の値は上書きできる

一度変数に格納した値は、後から上書きして変更することができます。





変数の設定	×	
{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		●変数が「TaxRate」になっている
変数: TaxRate {X}		のを確認
値: 0.10	[x] ()	●値に「0.10」と入力
		※これが上書きの処理
保存 キャン	セル	❼「保存」をクリック



では、このフローを実行して変数の値が上書きされることを確認してみましょう。

שלל ",□	Main ●「実行」をクリック
1	{x} 変数の設定 変数 TaxRate に値 0.08 を割り当てる
2	メッセージを表示 ↓ タイトルが である通知ポップアップ ウィンドウにメッセージ TaxRate を表示し、押されたボタンを ButtonPressed に保存します
3	{x} 変数の設定 変数 TaxRate に値 0.10 を割り当てる
4	メッセージを表示 夕イトルがである通知ポップアップウィンドウにメッセージ TaxRate を表示し、押されたボタンを ButtonPressed に保存します





(x) ButtonPressed OK

(x) TaxRate 0.10

1

② 変数を自動で作る方法

変数を作る2種類の方法のうち2つ目、つまり自動で作る方法について学習していきます。これ は特に決まった方法があるわけではなく、何か任意のアクションを作成すると自動的に変数が作ら れます。ここでは Excel を起動した際に自動的に作られる ExcelInstance 変数を確認してみましょ う。







Excel を閉じる	×	
✓ Excel インスタンスを閉じます 詳細		
パラメーターの選択	●続けて「Exc は「ドキュメン を選択	cel を閉じる前」 で ントを保存しない」
 全般 Excel インスタンス: %Excellnstance% 	~ 0	
Excel を閉じる前: ドキュメントを保存しない	~ ①	_
	●「保存」をク	リック
○ エラー発生時 保	キャンセル	

それでは作製したフローを実行してみましょう。空の Excel が開いて、そのまま閉じられます。

₀∕ ª שדעד–	\sim	Main	「実行」をクリック				
1		Excel の起動 プ 既存の Excel プロセス 存します	を使用して空白のドキュメントで Excel を起動し、Excel インスタンス ExcelInstance に保				
2		ビ Excelを閉じる ExcelInstance に	呆存されている Excel インスタンスを閉じる				

今回はただ単に Excel を起動して閉じるだけのあまり意味のないフローでしたが、「Excel の起動」のアクションを作成すると「ExcelInstance」という変数が自動で作成されることを確認したかったために行いました。

以上、変数を作る2種類の方法について学びました。

9.3 変数のデータ型について

変数にデータ(値)を格納すると、データの種類ごとに自動的に「型」が決められます。これを「変数 のデータ型」と呼びます。つまり箱の中身の種類のことです。例えば、箱の中身、つまり格納されたデー タが「1」や「2」などの数値なら計算に使えますが、「a」や「b」などのテキストなら計算には使えま せん。Power Automate for desktop でよく使うデータ型は下記の通りです。

データ型	特徴
数值型	0~9(マイナスも含む)といった数値を扱うデータ型。算術演算が可能。
テキスト型	あいう、abc、記号(?@など)といったテキスト(文字列)を扱うデータ型。
DateTime 型	1/19/2023、2:07:51 PM といった日付や時間を扱うデータ型。
ブール値型	True(真)とFalse(偽)を扱うデータ型。if文などの条件分岐によく利用されます。
リスト型	複数の値を1つの変数で管理できるデータ型。「仕切りのある箱」のイメージ。 仕切りで区切られた各空間に値が入っていて、全体を1つの箱と考えます。プロ グラミングにおける「配列」に相当します。
データテーブル型	複数の値を1つの変数で管理できるデータ型。リスト型と似ていますが、データ テーブル型は「仕切りの中にさらに仕切りのある箱」のイメージ。プログラミン グにおける「2次元配列」に相当します。
インスタンス型	Web ブラウザや Excel などアプリケーションを起動するアクションを作成した 際に自動生成される変数に対して適用されるデータ型。先ほどの例の ExcelInstance がまさにインスタンス型です。インスタンスとは、アプリケーシ ョンが起動して使える状態になっているもののことです。インスタンス型の変数 は、後のアクションで操作対象を指定するときに使います。
ファイル型	ファイルの情報を扱うデータ型。ファイルの名前やパス、拡張子などが格納され ています。
フォルダー型	フォルダーの情報を扱うデータ型。フォルダーの名前やパス、作成日などが格納 されています。

これだけではまだよくわからないと思いますので、この中でも特によく使う、数値型、テキスト型、 DateTime 型、リスト型、データテーブル型について、さらに詳しく学習していきましょう。

9.3.1 数値型

数値型とは、0~9(マイナスも含む)といった数値を扱うデータ型のことで、算術演算が可能です。フ ローデザイナーを起動し、次のように変数を設定し、フローを実行してみましょう。



変数の)設定	×
{x} 新規	視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 <u>詳細</u>	
変数: 値:	NewVar {x}	{x} (j)
	保存	ッンセル

※変数の設定アクションを追加しただけでは、変数に値は格納されません。フローを実行することで初めて変数へ値 が格納されますので、フローを実行しましょう

⑤ワークスペースに「変数の設定」アク ۲ ションが追加されていることを確認 o⁄° ا サブフロー 🗸 Main {x} 変数の設定 1 変数 NewVar に値 1 を割り当てる ⑥「実行」をクリック。 画面に大きな変化はありませんが、正常に実行されています。





これで先ほど変数に格納した値の「1」が数値型であることを確認できました。

また数値型では、数値そのものだけでなく	、算術演算を行った結果を変数に格納することもできます。
---------------------	-----------------------------

変数の設定	×	●「変数の設定」ダイアログボ
{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		ックスの値の欄に「%3+5%」 と入力
変数: NewVar {X}		
值: %3 + 5%	(**) 🕕	
		❷「保存」をクリック
保存 キャン	ッセル :	

先ほどと同様、フローを実行し、変数ペインの NewVar をダブルクリックして確認してみましょう。



以下、この後のテキスト型以降もこの要領でデータ型を確認していきます。

9.3.2 テキスト型

テキスト型とは、五十音の「あいう」、アルファベットの「abc」、記号の「?@」などといったテキスト(文字列)を扱うデータ型のことです。尚、ここからはワークスペースに「変数の設定」アクションを ドラッグアンドドロップした後の状態から解説します。

変数の	設定	×
{x} 新規	現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	5 <u>詳細</u>
変数: 値:	NewVar {x} 我々は宇宙人だ	{x} ①
	保存	キャンセル

フローを実行し、画面右側の変数ペインの NewVar をダブルクリックすると下図の「変数の値」画面が開かれます。



9.3.3 DateTime 型

DateTime型とは、1/19/2023、2:07:51 PM といった日付や時間を扱うデータ型のことです。

変数の設定	×	
{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		
変数: NewVar {x} 値: %d"2022-03-25"%	{x}] (j)	 「変数の設定」ダイアログボック スの値の欄に「%d"2022-03-25"%」 と入力。 ※2022 年 3 月 25 日の Datetime 型 を生成します。
保存 キャン	1211	●「保存」をクリック

フローを実行し、画面右側の変数ペインの NewVar をダブルクリックすると下図の「変数の値」画面が開かれます。



ここで、Datetime 型の変数の作るために「%d"2022-03-25"%」と入力することが、なぜわかったのか について説明します。それは公式ドキュメントで調べたからです。





●開かれた画面にて、キーボードの「Ctrl + F」を押すと、右上にページ内検索ボックスが現れるので「datetime」と入力し Enter キーを押します					
📑 変数のデータ型 - Power Automate 🗙 🕂					
← → C ☆ 🔒 learn.microsoft.com,	/ja-jp/power-automate/desktop-flows/variable-data-types	_			
<mark></mark> Microsoft Learn <u></u> ਨ≠⊐ ×	ント トレーニング 認定資格 Q&A コードサンブル ショー イ 	≺⊃ datetime	0/1	~ ~ ×	
Power Automate テンプレート♂ :	コネクタ 🗗 🛛 Power Automate について学ぶ 🗸			無料アカウン	
	Learn / Power Platform / Power Automate /		⊕ ⊕ :	≔ この記	
変数のデータ型	変数のデータ型			シンプルな	
変数のデータ型のプロパティ 変数と % 表記を使用する	[アーティクル]・2022/11/23・1 人の共同作成者		🖒 フィードバック	高度なデー インスタン	

→ C 🏠 🔒 learn.microsoft.com/ja-jp/p	ower-automate/desktop-f	lows/variable-data-types			Q	B
	日付と時間		datetime	1/1	^	~
タイトルでフィルター	• Datetime - 日作	すと時刻の情報が含まれていま	す。 変数の設定 アクション	ンを介して		
タ双にタ双(ヽ)ノ((6仕))⊘ ▲	日時変数を作成	なするには、次の式 %d"vvvv-N	1M-dd HH:mm:ss.ff+zzz"%	に入力パ		
変数のデータ型	ラメーターを入	、力します:				
変数のデータ型のプロパティ						
変数と % 表記を使用する	表記	内容				
UI 要素を使用して自動化する	VVVV	年				
画像を使用して自動化する	,,,,,					
デスクトップ フローでエラーを処理	MM	月				
する	dd	曜日				
デスクトップ フローを記録する	нн	時				
ループを使用する						
条件を使用する	mm	分間				
Web ページを自動化する	SS	秒				
カスタム フォームを作成する	ff	ミリ秒				
デスクトップ アプリケーションを自						
動化する	ZZZ	いにオノセット				
仮想デスクトップで自動化する	たとえば、 <mark>%d</mark> '	' 2022-03-25"% の場合、2022	年 3 月 25 日の日付をター	ゲット変数		
カフタムセレクターを作成する	に割り当てます					

⑦するとここに「%d"2022-03-25"%」の表記を見つけることができました。これを使えば良さそうだということがわかります。

※この文字列をコピーアンドペーストする場合、右端の「%」が、なぜか全角になっているの でこのままではエラーとなります。半角に修正してください。おそらく誤記と思われます。

このように、公式ドキュメントを使って、わからないことを調べることができます。少し読みづらいと ころもありますが、ひと通り基礎が身についた後は是非活用してみましょう。

さらに補足ですが、変数の設定ダイアログボックスにて「%d"2022-03-25"%」と入力した後にフロー を実行し、再び変数の設定ダイアログボックスを開いてみると、下図のように変更されています。

変数の設定	×
{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	
変数: NewVar {x} 临: %D'2022/03/25 00:00:00'%	「%d 2022-03-25 %」と入力したはすが「%D'2022/03/25 00:00:00'%」となっています。自動で変換されるようです。 {X} 「 そのため、始めから後者のように入力しても正しく動作します。
保存 キャン	ンセル

9.3.4 ブール値型

ブール値型とは、True(真)と False(偽)を扱うデータ型のことです。if 文などの条件分岐によく利用されます。例えば、アドレス一覧から未送信のアドレスにのみ、お知らせメールを送信したいといった場合に、送信済みを True、未送信を False と設定して条件分岐により挙動を変えることができます。ここではブール値型の簡単な確認だけを行います。

変数の)設定			×
{x} 新#	見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	とは以前作成した変数を上書きする 詳細	
変数: 値:	NewVar {X}			{x}] ()
12.				
			保存キ	ャンセル

フローを実行し、画面右側の変数ペインの NewVar をダブルクリックすると下図の「変数の値」画面が開かれます。

変数の値 NewVar (ブール値)	×	❸「ブール値」と表示されているのを確認 できます。これは、変数 NewVar はデータ 型が「ブール値型」であるという意味です。
True		
	閉じる	❹「閉じる」をクリック

9.3.5 リスト型

リスト型とは、複数の値を1つの変数で管理できるデータ型のことです。「仕切りのある箱」のイメージです。仕切りで区切られた各空間に値が入っていて、全体を1つの箱と考えます。プログラミングにおける「配列」に相当します。Power Automate for desktop では、Excel の列を管理する際によく利用されます。



リスト型の変数は2つの方法で作成することができます。1つ目は今までと同じ「変数の設定」アクシ ョンを使って作成する方法で、2つ目はリスト型の変数を作成するための「新しいリストの作成」という 専用のアクションを使って作成する方法です。イメージのしやすい「新しいリストの作成」アクションを 使った方法から解説します。まず、「空の仕切り箱」を作り、次に各仕切りの空間に値を格納していくと いった流れです。



◆「新しいリストの作成」アクションを使ってリスト型の変数を作成する方法

間に「0番目のアイテムです」という値を格納

④「項目をリストに追加」アクションを「新し」

いリストの作成」アクションの下にドラッグ

してみましょう。

♀Ξ リストのシャッフル

↓ リストを逆にする
 □ リストから重複する項...
 □ 共通リスト項目の検索
 □ リストの内容を減らす

🎦 データテーブル列をリ...

□ JSON をカスタムオブ…
 □ カスタムオブジェクトを…
 □ 項目をリストに追加
 + 新しいリストの作成

↓ リストの統合

項目をリストに追加	×	
ᆕ↓ 新しい項目をリストに追加 詳細	ſ	⑤「項目の追加」の欄に「0番
パラメーターの選択 > 全般		日のディナム(リ」と入力。ここは実際に格納する値を入力する箇所です。
項目の追加: 0番目のアイテムです	{x}] ()	❻「追加先リスト」の{x}マーク をクリック
追加先リスト:	{x} ()	2, , , , ,
<u>変</u> 数の検索 名前 型	211	
✓ 7□-変数 1 ↓ List リストー般の値		⑦「List」をクリックしてから 「選択」をクリック。「List」 は先ほど作成した空の仕切り 箱(変数)のことです。
選択	キャンセル	
項目をリストに追加	×	
━: 新しい項目をリストに追加 詳細		
パラメーターの選択 ~ 全般		
項目の追加: 0番目のアイテムです	{×} (j)	₿設定内容を確認し「保存」を クリック
追加先リスト: %List%	{x} ()	
保.t	₹ キャンセル]

これで仕切りのある箱、つまり変数%List%を作成し、その仕切りで区切られた空間の0番目に「0番 目のアイテムです」という値を格納することができました。 同様に「項目をリストに追加」アクションをあと2つ、ワークスペースにドラッグし「1番目のアイテムです」と「2番目のアイテムです」という値を格納します。

インデックス番号1番目のアイテムの設定画面です。



最後のインデックス番号2番目のアイテムの設定画面です。



	۲	م
∕° שלע [] ~	Main	●オベブの地中が中マオスレースの図
1	+ 新しいリストの作成 新しいリストを作成して List に保存	のようになっています。これで準備完 了です。
2	 —: 項目をリストに追加 * 項目 '0番目のアイテムです' をリスト List に追加 	
3	 —: 項目をリストに追加 * 項目 '1番目のアイテムです' をリスト List に追加 	❻「実行」をクリックして、データ型 がリスト型の変数をいよいよ生成して
4	 ・項目をリストに追加 ・ ・ ・	みましょう





ここで「List(リストテキスト値)」という表記についての補足です。これは、変数 List はデータ型 がリスト型で、各仕切り空間に格納されているアイテムはテキスト型であるという意味です。各仕切り 空間に格納されているアイテムが数値型なら「リスト数値」と表示されます。

このように少しややこしいのですが、リスト型のデータ型では、全体ではリスト型なのですが、各仕切 り空間に格納されているアイテムはテキスト型や数値型など、様々なデータ型のアイテムが格納可能な のです。

また、各仕切り空間に格納されているもののことを「値」「アイテム」「項目」と呼んできましたが、 どれも同じ意味です。 ここまで行った操作を図解すると下図のようになります。



◆「変数の設定」アクションを使ってリスト型の変数を作成する方法

次に、今まで学習した数値型やテキスト型などのデータ型を作成したときと同じ「変数の設定」アクションを使って、もっとシンプルにリスト型の変数を作成する方法を学習します。



変数の	設定			×	
{x} 新規	見や既存の変数に値を設定する、新しし	の変数を作成する、または以前作成した変	s数を上書きする <u>詳細</u>		❷変数名は「NewVar」の ままでもいいのですが 前
変数:	List {X}				回の学習に合わせ、ここで は「List」と入力しました
値:	%['0番目のアイテムです','1番目のフ	アイテムです','2番目のアイテムです']%	{x}	i	
		●値の欄は %['0番目のアイテムです と入力。各アイテムをシ ください。これがリスト	す','1 番目のアイティ シグルクォーテーシ 型の変数の 2 つ目の ⁻	ムです ョンで 作成方	', '2 番目のアイテムです ']% ´囲うのを忘れないようにして `法です。
			保存 キャンセル		●「保存」をクリック

	ÞI	۲		م
[_] שׂדַכם	~	Main $ imes$	⑤「実行」をクリック。画面に大きな変化は ありませんが、正常に実行されています。	
1		{x} 変数の設定 変数 List に値 ['0番目の	アイテムです', '1番目のアイテムです', '2番目のアイテムです'] を割り当てる	:

変数	×	{x}	
▶ 変数の検索		8	
◇ 入出力変数 0		5	
ここにはまだ入力または出力の変数がありません			
~ 7□-変数 1	7		●画面右側の変数ペインの「List」を ダブルクリック
(*) List [0番目のアイテムです,			



このように「変数の設定」アクションを使ってリスト型の変数を作成する場合、ダイアログボックスの 値の欄に下記のようにパーセント記号%と角カッコ[]を使って記述します。角カッコ[]の中には各仕切 り空間に格納したいアイテムを順番に記述します。先頭からインデックスが振られます。ここでもイン デックスが0から始まる点に注意です。

リスト型の作成方法 %[○○,△△,…,□□]%

また、今回のように仕切り空間に格納するアイテムがテキスト型の場合は、シングルクォーテーション (*) で囲う必要があります。数値型などテキスト型でなければシングルクォーテーションで囲う必要は ありません。例えば、%[10,100,1000]%、などです。

以上、2 つの方法でリスト型の変数を作成しましたが、今度は取得方法、すなわち作成したリスト型の 変数から各アイテム、つまり仕切り空間に格納されている値を取得する方法について学習します。先ほ どの続き、つまり「0 番目のアイテムです」などが設定された状態からスタートします。



メッセージを表示		×	
💭 メッセージ ボックスを表示します <u>詳細</u>			
パラメーターの選択			
~ 全般			▲今回けインデックス釆早0釆日の
メッセージ ボックスのタイトル:		{x} (i)	● プロページ デジス 留号 0 留日の アイテムを取得したいので「表示す るメッセージ」欄に%l ist[0]%と入
表示するメッセージ:	%List[0]%	[×] ①	カ。 カ。
			変数の選択の{x}マークをクリック してから入力するとミスを防げま す。
メッセージ ボックス アイコン:	เงงนั	~ Ū	
メッセージ ボックス ボタン:	ОК	~ Ū	
既定のボタン:	最初のボタン	~ Ū	•
メッセージ ボックスを常に手前に表示する	ō:		❸「メッセージボックスを常に手前 に表示する」をオンにします。図と
メッヤージ ボックスを自動的に閉じる:			同じ状態がオンです。
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,			
> 生成された変数 ButtonPressed			
◯ エラー発生時	保	存 キャンセル	❹「保存」をクリック
		lit.	
		٩	
₀⁄ª ザブフロー 〜 Main			❺「実行」をクリック
1 {x} <mark>変数の設定</mark> 変数 List に値['0番目のアイテムです', '1番目のアイテムです', '2番目のアイ	(テムです']を割り当てる	
×ッセージを表示 2 ↓ タイトルが である通 ButtonPressed	知ポップアップ ウィンドウにメッセージ List [0] を表示し、 打 に保存します	甲されたボタンを	
×			
0番日のアイティッオ	⑥ インデック	ス番号 0 番目のアイ]
CERW/ H/ACY	テムが表示さ	れているのを確認	
ОК	⑦「OK」をク	リック	

このように、リスト型の変数から格納されている値を取得する方法は下記のような書き方になります。

%変数名[インデックス番号]%



9.3.6 データテーブル型

データテーブル型とは、複数の値を 1 つの変数で管理できるデータ型のことです。リスト型と似てい ますが、データテーブル型は「仕切りの中にさらに仕切りのある箱」のイメージです。プログラミングに おける「2次元配列」に相当します。Power Automate for desktop では、「Excel を起動」アクション で自動生成される変数 ExcelInstance がデータテーブル型になります。つまり Excel の表全体(行と列) を管理するイメージです (列のみ管理する場合はリスト型となります)。Excel 操作を自動化する際に最 もよく使うデータ型なので、きちんと理解しておきましょう。「2次元配列」は少しイメージしづらいの で、下図のような流れで理解していきましょう。

2次元配列とは「仕切りの中にさらに仕切りのある箱」のことでしたね。つまり、リスト型を「入れ子」 にしたデータ型ということです。ですのでまず、リスト型から話を始めてみましょう。





ここで、この図が Excel の表(行と列)と似ていることにお気づきでしょうか。同じ範囲を選択状態 にしてみたものが下図です。「A、B、C」や「1、2、3」などのインデックス番号の振り方が異なるだけ で、考え方は同じであることがわかるかと思います。このように、データテーブル型は Excel と大変相 性の良いデータ型なのです。

	A1	0	· : ×	2 fx		
		A	В	C	D	E
0	1					
1	2					
2	3					
	4					
	5					
	6					
	-,					

今は2次元配列に何も値が格納されていない空の状態なので、適当に「あいう」「かきく」「さしす」 の値を格納してみて、さらに具体的にイメージしてみましょう。それから、この箱全体を表す変数名を 「%DataTable%」としておきましょう。



そして、この 2 次元配列から「あ」を取得したい場合、%DataTable[0][0]% と記述します。「い」なら %DataTable[0][1]%、「か」なら %DataTable[1][0]%、「し」なら %DataTable[2][1]% と記述します。つまり 2 次元配列から、格納されている値を取得する方法は下記のような書き方になります。

%変数名[行のインデックス番号][列のインデックス番号]%



これを Excel に対応づけると下図のようになります。

	Δ1		: x	√ fr	L a		
		0	1	2			
		А	В	С	D	E	
0	1	あ	い	ñ			
1	2	か	ぎ	<			
2	3	さ	L	す			
	4						
	5						
	6						
	~						

それでは、今度は実際に Power Automate for desktop 上でデータテーブル型の変数を作成してみま しょう。データテーブル型の変数を作成する方法は 2 つあります。1 つ目は、数値型やテキスト型の変数 を作成したときと同じ「変数の設定」アクションを使って作成する方法で、2 つ目は「新しいデータテー ブルを作成する」という専用のアクションを使って作成する方法です。まずは、イメージのしやすい後者 の専用のアクションを使った方法から学習していきましょう。

◆「新しいデータテーブルを作成する」アクションを使ってデータテーブル型の変数を作成する方法 白紙のフローデザイナーを起動した状態からスタートします。



データテーブ	ルを編集する 🛈			
0 Colu	ımn1 Column2	Column3	•	❺このように3行3列になって
1 2				いるのを確認します
Ð				
			保存 キャンセル	

データ	タテーブルを編れ	集する 🛈			
	Column1	Column2	Column3		▲4 に 4 列口ナカリッカーマに集団代比約にします。
0	あ {x}				●1111列日をクリックして編集可能状態にします。、 に「あ」と入力します。
1					
2					
				·	

データ	タテーブルを編	集する 🛈				
	Column1	Column2	Column3		Ð	
0	a	Li I	ð			
1	か	き	<			
2	č		j (x)			⑦同様に左図のようにすべてのセルに値を入力していきます
Ð						
				<u>保存</u> <u>+ヤン</u>	1211	❸「保存」をクリック

新しいデータ テーブルを作成する	×	③こちらでも3行3列になっているのを確認します
Ⅲ 新しいデータ テーブル変数を作成する <u>詳細</u>		
パラメーターの選択		
新しいテーブル 3 行, 3 列	編集	 ●自動生成された変数名は 「DataTable」です。今回は名
> 生成された変数 DataTable		前を変更せず、このまま使いま す。
保存	キャンセル	●「保存」をクリック



これでデータテーブル型の変数%DataTable%を作成し、その中に値を格納することで3行3列の2 次元配列を作成することができました。 次に、この2次元配列から特定の値、例えば「く」を取得し、表示してみましょう。「メッセージを表 示」アクションを利用します。





					ワークスペースに「メッセージを表 示」アクションが追加されました				
o⁄¤	" サブフロー 〜 Main		Main						
	1		田 新しいデータテーフ 新しいデータテーブル?	ルを作成する 2作成して DataTable に格納する		「●」実行」をクリック			
	2		メッセージを表示 タイトルが である通知 たボタンを ButtonF	ロポップアップウィンドウにメッセージ DataTable [1][2] を表示し、押され ressed に保存します					



今回は「く」を取得しましたが、他の値を取得したい場合の、変数に設定するインデックスと各値の関係は下図のようになります。例えば「し」を取得したいなら%DataTable[2][1]%となります。

変数0	変数の値								
Data	DataTable (Datatable)								
#	Column1		Co	lumn2	Co	Column3			
0	ぁ	[0][0]	ίı	[0][1]	õ	[0][2]			
1	か	[1][0]	き	[1][1]	<	[1][2]			
2	ź	[2][0]	L	[2][1]	す	[2][2]			

◆「変数の設定」アクションを使ってデータテーブル型の変数を作成する方法

次に、今まで学習した数値型やテキスト型などのデータ型を作成したときと同じ「変数の設定」アクションを使って、もっとシンプルにデータテーブル型の変数を作成する方法を学習します。



●全体を囲うのがパーセント記号と波カッコです ⇒ %{}%
 ●この中に3つの角カッコをカンマ区切りで並べます⇒ [],[],[]
 ●角カッコの中に値をカンマ区切りで並べます。今回はテキスト型の値なので各値
 をシングルクォーテーションで囲います ⇒ ['あ','い','う']

変数の	設定		×	
{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 <u>詳細</u>	
変数:	DataTable {x}		❷変数名は「NewVar」の のですが、前回の学習に では「DataTable」と入	Dままでもいい C合わせ、ここ 力しました
値:	%{['あ','い','う'],['か','き','く'],['さ','し','す']}?	6	{x} 〕	
	2 	〕値の欄に {['あ','い','う'],['か',' こ入力します。カンマとシング てください。これがデータテー	き',,'く'],['さ',,'し',,' ルクォーテーションの記述 -ブル型を作成するときの言	す']}% Š漏れに注意し 書き方です。
	④「保存	」をクリック 保	存 キャンセル	

	ÞI	۲		م
^ש לד	\sim	Main		
1		{x} 変数の設定 変数 DataTable に値 {['あ	5', 'い', 'う'], ['か', 'き', 'く'], ['さ', 'し', 'す'] } を割り当てる	:

変数 ×	{x}	
▶ 変数の検索	8	
◇ 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		▲西五ナ側の亦粉ペインにたて「DataTable」た
		●画面石闸の友女ペインにある「Dururuble」を ダブルクリック
∨ 7□−変数 1		
(*) DataTable 3 行, 3 列		

変数(の値								\times
Data	Table (Datat	able)							
#	Column1	Column2	Column3						
0	あ	()	ð						
1	か	き	<	[1 65 ペーシ	ジと同じ結果になりまし	した		
2	ż	L	す	L					
						❸「閉じる」をクリ	ック	閉じる	

このように、データテーブル型(2次元配列)は少し複雑ですが、しっかりと使いこなせるようになり ましょう。

10 繰り返し処理

前のセクションでは、リスト型やデータテーブル型の変数の中から格納されている値を取得したい際 に、角カッコ[]とインデックスを使って「%List[0]%」のように記述して値を取得しました。では、格納 されている値がたくさん存在し、すべての値を取得したい場合、リスト型なら「%List[0]%、%List [1]%、%List[2]%、・・、%List[100]%」、データテーブル型なら「%DataTable[0][0]%、%DataTa ble[0][1]%、%DataTable[0][2]%、…、%DataTable[100][100]%」といった具合に、すべての数字を記 述して取得するのはいかにも面倒なことがわかります。このようなときに繰り返し処理を使うと、少な い記述で済み、汎用性も高まります。Power Automate for desktop では、繰り返し処理を「反復」や 「ループ」と呼び、下図のようにフローデザイナーにループアクショングループが用意されています。



「For each」アクションは、リスト型やデータテーブル型のアイテムがなくなるまで繰り返し処理を 行い、「Loop」アクションは、指定した回数分だけ繰り返し処理を行います。アイテム数が定まらない ときは繰り返し回数を指定しなくて済む「For each」を使い、アイテム数が決まっているときは回数を 指定する「Loop」を使うといった具合に使い分けます。厳密に言うと、両方とも全く同じことができる のでどちらを使うかは好みで決めて構いません。この説明だけではまだよくわからないと思いますので、 具体的に使ってみましょう。まずはシンプルでわかりやすい Loop アクションから操作してみましょう。

10.1 Loop アクション

Loop アクションでは、繰り返しを管理するための変数 LoopIndex が自動的に生成され「開始値」に 設定された値が初期値として格納されます。その後、変数 LoopIndex は繰り返し毎に「増分」に設定さ れた値の分だけ変化し、「終了」に設定された値に達すると繰り返し処理が終了します。



Loop アクションと End アクションの間に繰り返しを行いたい処理を配置します。そのため、ここでは「メッセージを表示」アクションを配置して、先ほど設定した「開始値」から「終了」まで「増分」に 従って変化する変数 LoopIndex を繰り返し表示してみましょう。






		م	
₀⁄º サブフロ- `	Main		♥'実行」をクリック
1	~ (Loop ステップ 1 を使用して、 LoopIndex を 1 から 3 までループ処理します		
2	メッセージを表示 夕イトルが である通知ボップアップ ウィンドウにメッセージ LoopIndex を表示し、押されたボタンを ButtonPressed に保存します	÷	
3	F ^D End		





このように変数 LoopIndex は 1 回目のループで 1、2 回目のループで 2、3 回目のループで 3 となり 終了します。増え方は増分に設定した「1」ずつ増えていきます。

Loop	LoopIndex	
1回目	1 .	増分 ⊥1
2回目	2	τı
3回目	3	+1

◆Loop アクションを利用してリスト型の変数からすべての値を取り出す

次に、リスト型の変数から、格納されているすべての値を取り出す操作を Loop アクションを利用して 実行してみましょう。









このように月曜から日曜までのメッセージが順番に表示されました。ここで、この Loop の原理について説明します。



Loop	×
◯ 指定された数のアクション ブロックを反復	処理します <u>詳細</u>
パラメーターの選択 開始値: 0 終了: 6	次に、変数%NewVar%のインデックス 0~6 に対応した変数%LoopIndex%を作成し ました。 つまり「開始値」を 0、「終了」を 6、「増分」を 1 とした値が、Loop が回る毎に% LoopIndex%に順番に格納されていくように設定しました。
増分: 1 > 生成された変数 LoopIndex	{x}] () 保存 キャンセル

今回の Loop を図解すると下記のようなイメージとなります



%LoopIndex%	%NewVar[LoopIndex]%	取得される値							
0	%NewVar[0]%	月							
1	%NewVar[1]%	火	月	火	水	木	金	土	E
•	•	•		1	2	3	4	5	6
6	%NewVar[6]%	日				6New0/ar%			

◆Excel における Loop アクションの使用例

もう少し実用的な例として Excel と Loop アクションの組み合わせについて学習します。前回作成した「月~日」までの曜日を Excel に転記してみましょう。



次にフローデザイナーを開きリスト型の変数を作成します



変数の設定	×	
 {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 		₿変数名は NewVar をそのまま
変数: NewVar {x}		変更せず使います
値: %['月', '火', '木', '金', '土', '日']%	{x}	❹値はリスト型にし、各アイテム
		として曜日を入力しましょう。各 アイテム自体はテキスト型なの
		でシングルクォーテーションで 囲みましょう。
保存 キャン	ンセル	┓╹ぽ~









で LoopIndex を利用します











実行が終了するまでしばらく時間がかかりますので待ちましょう。また実行中はどこも操作しないようにしましょう。 エラー表示なく終了すれば OK です。それでは、デスクトップにある myloop.xlsx を開いてみましょう。





ここで Loop アクションにより生成される各値を整理すると下記のようになります。変数%LoopIndex%が0~6まで変化することをうまく利用して他の値も変化させています。前回の例では 各曜日をメッセージボックスに表示していましたが、今回はそれが Excel への転記に置き換わっただけ という違いですので、理解があいまいな場合は前回の例をもう一度復習してみてください。

%LoopIndex%	%NewVar[LoopIndex]%	列	行 (%LoopIndex+1%)	書き込まれる値
0	%NewVar[0]%	A	1	月
1	%NewVar[1]%	А	2	火
2	%NewVar[2]%	А	3	水
3	%NewVar[3]%	А	4	木
4	%NewVar[4]%	А	5	金
5	%NewVar[5]%	А	6	土
6	%NewVar[6]%	А	7	日



%NewVar%

10.2 For each アクション

前回までは、繰り返し処理のうち「Loop」アクションについて学習しました。このセクションでは、 もう一つの繰り返し処理である「For each」アクションについて学習していきます。2 つの違いについ てもう一度おさらいすると、「Loop」アクションは、指定した回数分だけ繰り返し処理を行うのち対し、 「For each」アクションは、回数を指定せずにアイテム数分の繰り返し処理を行います。ひと言で言う と「Loop は指定した回数だけやってくれる」、「For each はアイテムがなくなるまで、あるだけやっ てくれる」という違いです。アイテム数が決まっているときは「Loop」を使い、アイテム数が定まらな いときは「For each」を使うといった具合に使い分けます。厳密に言うと、両方とも全く同じことがで きるのでどちらを使うかは好みで決めて構いません。

「Loop」アクションでは、繰り返しを管理するために LoopIndex という変数が自動生成されました。 この変数 LoopIndex に「開始値」「終了」「増分」の値を設定することで繰り返しの回数を指定したの でした。一方「For each」アクションでは CurrentItem という変数が自動生成されます。繰り返しの回 数を指定しないため、この変数 CurrentItem には「現在抜き取られている値」が繰り返し処理の回る毎 に順番に格納されていきます。

それでは実際に「For each」アクションを操作してみましょう。前のセクションと同じように「月~ 日」までの曜日を順番にメッセージボックスに表示するフローを作成していきます。





アクション		۲	
🔎 アクションの検索	₀⁄º サブフロ- ∨	Main	
> 変数	1	{x} 変数の設定 _{変数} NewVar に値 ['月', '火'	, '水', '木', '金', '土', '日'] を割り当てる
> 条件 ✓ ル <u>−プ</u>		🗘 For each	
G For each			❺「For each」アクションを 「変数の設定」の下へドラッグ



アクション		۲	
🔎 アクションの検索	₀⁄° サブフロ- ∨	Main	
> データベース	1	{x} 変数の設定 変数 NewVar に値 ['月', '火', '水', '木', '金', '土', '日'] を割り当	578
> メール > Exchange Server	2	♥ For each CurrentItem in NewVar : ■ メッセージを表示	
> Outlook ✓ メッセージボックス ↓ メッセージを表示	3	P End :	●「メッセージを表示」ア クションを「For each」と 「End」の間にドラッグ



2	✓ ∅ For each CurrentItem in NewVar
3	メッセージを表示 ↓ タイトルが である通知ポップアップ ウィンドウにメッセージ '今日は' Currentitem '曜日です' を表示し、押 されたボタンを ButtonPressed に保存します
4	P End

1回目の Loop





このように月曜から日曜までのメッセージが順番に表示されました。ここで、この For each の原理について説明します。



◆Excel における For each アクションの使用例

Loop アクションのときと同様に「月~日」までの曜日を For each アクションを使って Excel に転記 してみましょう

●まずは、今回のフローの操作対象となる空の Excel ドキュメントとし て、デスクトップにある先ほどと同じ「myloop.xlsx」を使います。そし て、このドキュメントは一度閉じておいてください。 \bullet : \times \checkmark f_x A1 X ≣ В С D Α 1 myloop.xlsx 2 3 4 5 6

> 7 8

次にフローデザイナーを開きリスト型の変数を作成します











For each ×	❶「反復処理を行う値」に 先ほど作成した変数%N
○ 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 ブロックを繰り返して実行します <u>詳細</u>	ewVar%を指定。 右横の{x}マークの「変数
	の選択」をクリックして
パラメーターの選択	から選択すると入力ミスを防げます
反復処理を行う值: %NewVar% {X} ()	8 M 1 4 9 0
保存先: CurrentItem {x}	1 変数 CurrentItem が 自動ブ作成されます。
	れに%NewVar%の中か
	ら現在抜き取られている
	値が順笛に恰納されてい きます。
	❹「保存」をクリック



Excel ワークシートに書き込む ×	●書き込む対象先として先ほど 記動した Excal ドキュメントを表	
曜 目 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 <u>詳細</u>	起動した Excel Instance%を指定	
パラメーターの選択 ~ 全般	●書き込む値には、変数%NewVa r%に格納されている各値(曜日) のうち、現在抜き取られている値 を表す%CurrentItem%を指定	
Excel 1 ノスタクス: %Excellinstance% 「 書き込む値: %CurrentItem% {X} ()	●「書き込みモード」は「指定したセル上」を選択	
書き込みモード: 指定したセル上 列: A	 「列」にはAと入力。Excelの A1 セルからA7 セルに「月~日」 の曜日を書き込みたいため。 	
آت: (א) (ب)	②ここで「行」に指定できる値がないことに気づきます。	
○ エラー発生時 保存 キャンセル	⑦そのため、とりあえず「キャンセル」をクリック	

手順20にて「行」に何を指定すればいいのかわからないという問題が発生しました。ExcelのセルA1 ~A7 に値を書き込みたいので、繰り返しの度に行番号が「1~7」に変化してくれる値が欲しいですね。 ちなみに Loop アクションのときは変数%LoopIndex%がこの役割を担ってくれていました。下図のよ うに設定したことを思い出してみましょう。



この%LoopIndex%に相当する変数が For each アクションにもあればいいのですが、残念ながらあり ません。For each アクションの変数%CurrentItem%には変数%NewVar%の「月~日」の各値そのもの が順番に格納されるため、数値の 1~7 を設定できる機能はありません。そのため、自分で作るしかない のです。それでは「変数の設定」アクションを使って、%LoopIndex+1%に相当する変数、つまり「今 何回目の繰り返しか?」を管理する変数を作りましょう。



92 / 116



このままでは変数%RowCount%は 1 のままです。繰り返し処理が回る毎に 1 ずつ増やす必要があります。「変数を 大きくする」アクションを使って%RowCount%を増やしましょう。



「変数を大きくする」アクションのダイアログボックスが表示されま うに設定していきましょう	もしたので下図のよ	
変数を大きくする	×	
~7 変数の値を一定量大きくする <u>詳細</u>	 ①値を大きくす 亦数タに%Pow 	る対象の Count%
パラメーターの選択	を指定	
✓ 全般 変数名: %RowCount% 大きくする数値: 1	{x} ① {x} ① (x) ① ① ① ① ①	するのか ろです。 したいの します。
保存	キャンセル 167 (保存」をク	リック



Excel を閉じる		×	
✓ Excel インスタンスを閉じます 書	<u> </u>		
パラメーターの選択 ~ 全般			●閉じる対象として現在操作している Excel ドキュメントを表す変数%ExcelInstance%を指定
Excel インスタンス: %Excellns	stance%	✓ ①	
Excel を閉じる前: ドキュメント	トを保存	~ 0	
			■ 「ドキュメントを保存」を選択
◯ エラー発生時		保存 キャンセル	●「保存」をクリック



実行が終了するまでしばらく時間がかかりますので待ちましょう。また実行中はどこも操作しないようにしましょう。 エラー表示なく終了すれば OK です。それでは、デスクトップにある myloop.xlsx を開いてみましょう。



ここで For each アクションにより生成される各値を整理すると下記のようになります。行番号を管理 する変数%RowCount%を用意し 1 ずつ増やすアクションを追加したことが今回のポイントです。ま た、%RowCount%の終了値「7」を指定する必要がない点にも注目です。なぜならば、For each アクシ ョンは反復処理を行う値である%NewVar%の中身(%CurrentItem%)がなくなるまで処理を行ってく れるからです。

回数	%CurrentItem%	列	行 (%RowCount%)	書き込まれるセル	書き込まれる値
1回目	月	А	1	A1	月
2 回目	火	А	2	A2	火
3回目	水	А	3	A3	水
4回目	木	А	4	A4	木
5 回目	金	А	5	A5	金
6回目	±	А	6	A6	±
7 回目	日	А	7	Α7	B

書き込み先のセル A1~A7 の行番号「1~7」を管理する変数%RowCount%は、初期値に 1 を格納した後、繰り返し処 理が回る毎に 1 ずつ増えていきます。この書き込み対象のセルに、変数%NewVar%の各値(月~日)のうち、現在抜 き取られている値を表す変数%CurrentItem%の中身が順番に書き込まれていきます。



11 条件分岐

「金額が1万円以上のときのみ実行」「値が空のときはスキップ」「1年以内に購入履歴のある顧客にのみメールを送信」など、条件によって処理内容を変えることを「条件分岐」と呼びます。Power Automate for desktop では、下図のように条件分岐を行うためのアクションである「条件グループ」が 用意されています。

アクション		۲		
🔎 アクションの検索	_ם ∕° שלע שע− –	Main		
> 変数 > 多件				
⊟⊋ Case				
T→ Else	フロー す。こ	-デザイナーのアクショ この中でも「if」「Else	ンとして「ś if」「Else	条件グループ」が用意されていま 」をよく使います。
⊣L; Else if				

11.1 if アクション

それでは条件分岐の「if」「Else if」「Else」アクションを使って簡単なフローを作成してみましょう。 メッセージボックスを表示し、「はい」をクリックした場合と「いいえ」をクリックした場合、さらに「キ ャンセル」をクリックした場合で処理を変えるというシンプルなフローです。まずは if アクションから です。







99 / 116

If	×	※「オペランド」と「演算子」 については後ほど解説します
↓ このステートメントで指定	こした条件を満たす場合に実行する、アクション ブロックの開始を示します <u>詳細</u>	
パラメーターの選択		9 「最初のオペランド」の欄 に%ButtonPressed%と入力
最初のオペラント: 演算子:	%ButtonPressed% {X} (i) と等しい (=)	❶「演算子」の欄は「と等しい
2 番目のオペランド:	Yes {X} ()	(=)」を選択
		❶「2 番目のオペランド」に欄 は「Yes」と入力
	保存 キャンセル	■ ⑫「保存」をクリック





「はい」がクリックされたとき、つまり変数%ButtonPressed%に Yes が格納されたときの処理を設定していきます				
メッセージを表示			×	
「,」 メッセージ ボックスを表示します <u>詳細</u>				
パラメーターの選択				
◇ 全般				
メッセージ ボックスのタイトル:		{x}	0	
表示するメッセージ:	「はい」がクリックされました	{x}	-	●「表示するメッヤージ」の欄に
				『「はい」がクリックされました』 と入力
メッセージ ボックス アイコン:	เมงร์	~] 🕡	
メッセージ ボックス ボタン:	ОК	~] 🖸	・ 「メッセージボックスボタン」 の欄は「OK」を選択
既定のボタン:	最初のボタン	~] ()	
メッセージ ボックスを常に手前に表示する			•	●「メッセージボックスを常に手前に表示する」を図のようにオン
メッセージ ボックスを自動的に閉じる:			()	にします
> 生成された変数 ButtonPressed2				
○ エラー発生時		保存 キャン	セル	●「保存」をクリック

ここで一度実	行してみましょう	
	٩ ا	
₀⁄º サブフロ- 丶	Main	
1	メッセージを表示 ♀ タイトルが である通知ポップアップ ウィンドウにメッセージ '下記いずれかのボタンとクリックしてください' を表示し、押されたボタンを ButtonPressed に保存します	「はい」がクリッ クされたとき、つ まり変数%Butto nPressed%='Ye
2	✓ ↓ If ButtonPressed ='Yes' then	s'のときに、この 中のアクション
3	メッセージを表示 ↓ タイトルが である通知ポップアップ ウィンドウにメッセージ '「はい」がクリックされました' を表示 し、押されたボタンを ButtonPressed2 に保存します	が実行されるこ とを確認してい きます
4	P End	



11.2 Else アクション

これで「if」アクションが正しく動作することを確認できました。しかし、手順 20 にて「いいえ」を クリックしても何も起きません。「いいえ」をクリックしたときにもメッセージを表示させたいもので す。ここで「Else」アクションの登場です。











ちなみに手順9にて「はい」をクリックすると先ほどと同様『「はい」がクリックされました』と表示 されます。つまり if アクションのみが実行され、Else アクションは実行されません。このように条件を 満たした時点で他の条件判定は行われないという点は重要なので覚えておきましょう。

11.3 Else if アクション

そして、さらに条件を増やしたい場合には「Else if」アクションを使います。まずは次の図に従って「はい」「いいえ」に加え「キャンセル」ボタンを追加しましょう。







Else if	×	
ーに、前の If ステートメントで打 アクション ブロックの開対	皆定した条件を満たしていないが、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、 台を示します <mark>詳細</mark>	
パラメーターの選択		 ⑤「最初のオペランド」の 欄に%ButtonPressed% と入力
最初のオペラント: 演算子:	%ButtonPressed% {X} (i) と等しい (=)	❻「演算子」の欄は「と等しい(=)」を選択
2 番目のオペランド:	No {X} (i)	⑦「2番目のオペランド」 欄に「No」と入力
	保存 キャンセル	❸「保存」をクリック

	٩ ٩	
₀⁄º サブフロ- 〜	Main	
1	メッセージを表示 ↓ タイトルが である通知ポップアップ ウインドウにメッセージ '下記いずれかのボタンをクリックしてください' を表示し、押されたボタンを ButtonPressed に保存します	
2	$\sim \downarrow$ If ButtonPressed ='Yes' then	
3	メッセージを表示 ↓ タイトルが である通知ポップアップ ウィンドウにメッセージ '「はい」がクリックされました' を表示 し、押されたボタンを ButtonPressed2 に保存します	
4	ー G Else if ButtonPressed ='No' then メッセージを表示	♀ 「いいえ」がクリックさ
5		れたときの「メッセージ を表示」アクションを「E
6	メッセージを表示 ✓ タイトルが である通知ポップアップ ウィンドウにメッセージ 「いいえ」がクリックされました を表示 し、押されたボタンを ButtonPressed3 に保存します	Ise It」と「Else」の間 にドラッグして移動
7	P End	

ここで現在の状況を確認しておきます。ここまでは「いいえ」がクリックされたときの「メッセージを表示」アクションを「Else if」の中に移動して見やすいように整理しただけです。また「いいえ」をクリックすると変数%ButtonPressed%には「No」が格納されますので、これを「Else if」の条件式に利用しています。












11.4 「if」「Else if」「Else」の動作イメージ

「if」「Else if」「Else」について、まだよくわからないかもしれませんので、下記のようにイメージ してみてください。

if 条件式1
if アクションの条件式1を満たしたときのみ、ここのアクションが実行されます。
Else if 条件式 2
条件式1を満たさず、かつ条件式2を満たしたときのみ、ここのアクションが実行されます。
Else if 条件式 3
条件式1と2を満たさず、かつ条件式3を満たしたときのみ、ここのアクションが実行されます。
 ※Else if はいくつでも追加できます。さらに、なくてもOKです。
Else

Else は条件式が不要です。

11.5 オペランドと演算子

if アクションの設定ダイアログボックスにて、下図のように「オペランド」と「演算子」という言葉が 出てきました。

If		×
↓ このステートメントで指知	宅した条件を満たす場合に実行する、アクション ブロックの開始を示します <u>詳細</u>	
パラメーターの選択		
最初のオペランド:	%ButtonPressed%	{x}
演算子:	と等しい (=)	~ (i
2 番目のオペランド:	Yes	{x} (i)
	保存	キャンセル

オペランドとは「被演算子」つまり演算の対象となる変数や値のことです。例えば「X=100」のオペラ ンドは「X」と「100」です。また「=」は「演算子」と呼ばれます。上図に対応付けると「最初のオペラ ンド」は「X」、「2番目のオペランド」は「100」、演算子は「=」となります。「Y>35」であれば「Y」 と「35」がオペランドで「>」が演算子、さらに上図に対応付けると「最初のオペランド」は「Y」、「2 番目のオペランド」は「35」、演算子は「>」となります。



演算子	演算子	
と等しい(=)	次を含まない	
と等しくない(<>)	空である	
より大きい(>)	空でない	
以上である(>=)	先頭	>
より小さい(<)	先頭が次でない	
以下である(<=)	末尾	
次を含む	末尾が次でない	

上図の if アクションの設定ダイ アログボックスの下矢印「V」マ ークをクリックすると出てくる各 演算子 11.6「メッセージを表示」アクションで変数に格納される値について

先ほどの if アクションの例では「はい」「いいえ」「キャンセル」のうち、クリックしたボタンに応 じて表示するメッセージを振り分けました。

メッセージを表示		×	
💭 メッセージ ボックスを表示します <u>詳細</u>			
パラメーターの選択			
~ 全般			
メッセージ ボックスのタイトル:		{x} ()	
表示するメッセージ:	下記いずれかのボタンをクリックしてください	{x} (j)	
メッセージ ボックス アイコン:	質問	~ i	「メッセージボックスボタン」
メッセージ ボックス ボタン:	はい - いいえ - キャンセル		の下矢印「∨」をクリックする ことで、表示させるボタンの種
既定のボタン:	最初のボタン	~ i	類を選択できるのでした
メッセージ ボックスを常に手前に表示する:		(i)	
メッセージ ボックスを自動的に閉じる:		(i)	「はい」「いいえ」「キャンセ
> 生成された変数 ButtonPressed			ル」のうら、クリックしたもの に応じて、それぞれ「Yes」「N
◯ エラ-発生時			o」 Cancer」という個が変数 ButtonPressed に格納される のブレた
	保存	キャンセル	

実は「メッセージボックスボタン」の選択項目は下記のように全部で6つあります。

メッセージ ボックス ボタン:	はい - いいえ - キャンセル 🗸 🛈		
選択項目			
ОК			
OK -キャンセル	×		
はい – いいえ	ア記いずれかのボタンをクリックしてく ださい		
はい - いいえ - キャンセ			
中止 - 再試行 - 無視	(はい) いいえ キャンセル		
再試行 - キャンセル			

また「はい」「いいえ」などの各ボタンをクリックしたときに変数%ButtonPressed%に格納される値 も変化します。変数に格納された値は、後のフローで使用することができます。if アクションでは、条件 判定に使いました。メッセージボックスボタンと変数%ButtonPressed%に格納される値との関係は下 図の通りです。

選択項目	イッセージギックフギタン	ボタンクリック時に変数に格納される値	
		クリックしたボタン	格納される値
ОК	ОК	ОК	ок
OK -キャンセル	OK キャンセル	ОК	ОК
		キャンセル	Cancel
はい – いいえ	(はい) いいえ	はい	Yes
		いいえ	No
		はい	Yes
はい – いいえ – キャンセル	(はい) いいえ キャンセル	いいえ	No
		キャンセル	Cancel
中止 - 再試行 - 無視		中止	Abort
	中止 再試行 無視	再試行	Retry
		無視	Ignore
再試行 - キャンセル	再試行キャンセル	再試行	Retry
		キャンセル	Cancel

12 作成したフローを共有する方法

作成したフローをチーム内で共有したいと思ったとき、有料のクラウド版であればクラウド上で簡単 に共有できますが、無料のデスクトップフローではこの方法を使えません。ですが、フローの内容をコピ ペしたファイルを作成し、このファイルをメールに添付するなどして共有することができます。ここで は無料のデスクトップフローをコピペにより共有する方法について解説します。例として、前のセクシ ョンで作成した「はい」「いいえ」「キャンセル」のifアクションのフローを使います。

フローを送る側の手順 Q <mark>。/</mark>9 サブフロー 🗸 Main メッセージを表示 タイレーンをある。 タイトルがである通知ボッブアップウィンドウにメッセージ・下記いずれかのボタンをクリックしてください、を表示し、押 されたボタンを ButtonPressed に保存します ●キーボードの「Ctrl」を押し 1 ながら「A」を押して(以下「C trl + A」のように記載)ワー ✓ ↓ If ButtonPressed ='Yes' then : 2 クスペースにあるアクション すべてを選択状態にします メッセージを表示 メリセーンをおか タイトルがである通知ポップアップウィンドウにメッセージ「はい」がクリックされました。を表示し、押されたボタ ンを ButtonPressed2 に保存します 3 ✓ -☐ Else if ButtonPressed ='No' then : 4 メッセージを表示 タイレーン さない タイトルが である通知ポップアップ ウィンドウにメッセージ 「いいえ」がクリックされました'を表示し、押されたポ タンを ButtonPressed3 に保存します 5 ②続けて「Ctrl + C」でコピー ∽ ⊑→ Else 🗄 6 メッセージを表示 メリセーンをおか タイトルがである通知ボッブアップウィンドウにメッセージ「「キャンセル」がクリックされました。を表示し、押され たボタンを ButtonPressed4 に保存します 7 P End : 8

・メモ帳アプリ」などテキストエディタを開きます	④「Ctrl + V」で貼り付けた後、名前を付けてファイルを保存します。こ こでは「条件分岐フロー.txt」と名付けました。
⑦ 無題 - メモ帳 ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)	● 条件分岐フロー・txt - メモ帳 ファイル(F) 編集(E) 書式(O) 表示(V) ヘルブ(H) Display.ShowMessageDialog.ShowMessage Message: \$'``下記いずれかのボタンをクリ・ Display.Icon.Question Buttons: Display.Buttons.YesNoCancel DefaultButton: Display.IsonMessageDialog.ShowMessage Message: \$'``「「はい」がクリックされ: Display.ShowMessageDialog.ShowMessage Message: \$'``「はい」がクリックされ: Display.Icon.None Buttons: Display.Buttons.OK DefaultButton: Display.DefaultBu ButtonPressed = \$'``No`` THEN Display.ShowMessageDialog.ShowMessage Message: \$'``「いいえ」がクリックされ Display.ShowMessageDialog.ShowMessage Message: \$'``「いいえ」がクリックされ ButtonPressed = \$'`No`` THEN Display.Icon.None Buttons: Display.Buttons.OK DefaultButton: Display.DefaultBu ButtonPressed = \$'`No`` THEN Display.Icon.None Buttons: Display.Buttons.OK DefaultButton: Display.DefaultBu ButtonPressed=> ButtonPressed3 ELSE Display.ShowMessageDialog.ShowMessage Message: \$'``「キャンセル」がクリック Display.Icon.None Buttons: Display.Buttons.OK DefaultButton: Display.DefaultBu ButtonPressed=> ButtonPressed4 END

このテキストファイル「条件分岐フロー.txt」をメールや共有ストレージなどを使って共有したい人 に渡します。そして次は、受け取った側の人の手順です。

フローを受け取った側の手順

////////////////////////////////////		
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルブ(H) Display.ShowMessageDialog.ShowMessage Mess Display.Icon.Question Buttons: Display.But	age: \$'''下記いずれかのボタンをクリ ton <mark>s.YesNoCancel DefaultButton: Dis</mark>	
Is topMost: True ButtonPressed=> ButtonPress IF ButtonPressed = \$'''Yes''' THEN Display.ShowMessageDialog.ShowMessage Display.Icon.None Buttons: Display.Buttons ButtonPressed=> ButtonPressed2 ELSE IF ButtonPressed = \$'''No''' THEN Display.ShowMessageDialog.ShowMessage	sed Message: \$'``「はい」がクリックされ、 .OK DefaultButton: Display.DefaultB Message: \$'``「いいえ」がクリックさ	t取った「条件分岐フロ xt」を開き「Ctrl + A」 選択し「Ctrl + C」でコ します
Display.Icon.None Buttons: Display.Buttons ButtonPressed=> ButtonPressed3 ELSE Display.ShowMessageDialog.ShowMessage Display.Icon.None Buttons: Display.Buttons	.0K DefaultButton: Display.DefaultBu Message: \$'''「キャンセル」がクリッ .0K DefaultButton: Display.DefaultBu	
ButtonPressed=> ButtonPressed4 END		
ファイル 編集 デバッグ ツール 表示 ^		
	✓ Main	
 > 変数 > 条件 	●	Kの ノローテサイテーを 、ワークスペースの何も ところをクリック
> ループ		
	٩	
_ס /ª ザブフロー 〜 Main X		
メッセージを表示 1 タイトルが である通知ボッブアップ ウィンドウにメッイ 押されたボタンを ButtonPressed に保存します	セージ '下記いずれかのボタンをクリックしてください' を表示し、	
2 ∽ ↓ If ButtonPressed ='Yes' then :	● 「C ● す	Xtrl + V」で貼り付けま
メッセージを表示 3 ↓ タイトルが である通知ポップアップ ウィンドウト タンを ButtonPressed2 に保存します	にメッセージ 「「はい」がクリックされました [、] を表示し、押されたボ :	
4		
	にメッセージ ' 「いいえ」がクリックされました' を表示し、押された :	
6 ∽ r Else :		
Xッセージを表示 ア ダイトルが である通知ポップアップ ウィンドウト れたボタンを ButtonPressed4 に保存しま	にメッセージ 「「キャンセル」がクリックされました'を表示し、押さまます	で無事フローを共有する
8 🏳 End 🗄	28	かしさました

手順 4 にてメモ帳アプリにフローを貼り付けたとき、英語がズラッと並んでいるのを確認できます。 これは Power Automate for desktop のフローの本体である「Robin(ロビン)」というプログラミン グ言語です。今まではアクションをワークスペースへドラッグしてフローを組み立てていましたが、実 は裏側で Robin へと変換されていたのです。今は Robin を読み書きすることはあまりないかもしれませ んが、将来的にもっと細かいフローを作成したいと思ったときは、是非 Robin の学習にチャレンジして みてください。

13 最後に

これにて「Microsoft Power Automate for desktop 活用講座 ①基本操作編」は終了です。何かお気 づきの点がございましたら「繊細無業おじさんパソコン IT 学習会」へお問い合わせください。この後は 「メール自動一斉送信編」を公開予定です。

お問い合わせ先: smopclearning@gmail.com