



また働く！ 働いている今を バージョンアップする！

第4回

RPAの入り口を見る

主催

鎌倉市商工課・かまくら主婦's ネットワーク



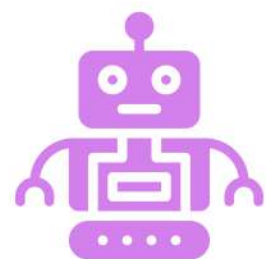


ロボティック

プロセス

オートメーション

RPA (Robotic Process Automation) とは？

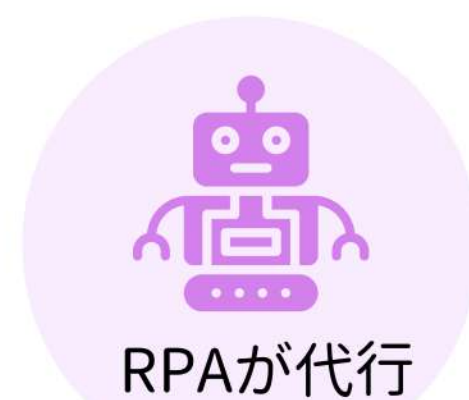


パソコンで行うデスクワーク（主に定型作業）を、
パソコンの中にあるソフトウェア型のロボットが代行・自動化する仕組み

BEFORE パソコンでの煩雑なデスクワーク



AFTER RPAで自動化



- 複数のアプリケーションにまたがった自動化ができる
- 既存のアプリケーションの変更が不要
- 開発速度が速い
(積み木遊び感覚で作成できる)

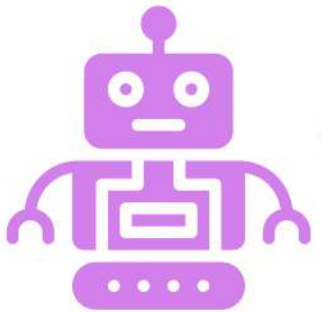




RPAシナリオ構築に必要なスキル

- 1 プログラミングの基礎知識
- 2 動かすシステムの作りや操作に習熟
- 3 業務の分解（＝マニュアル作成）
- 4 発想力
- 5 RPAの使い方

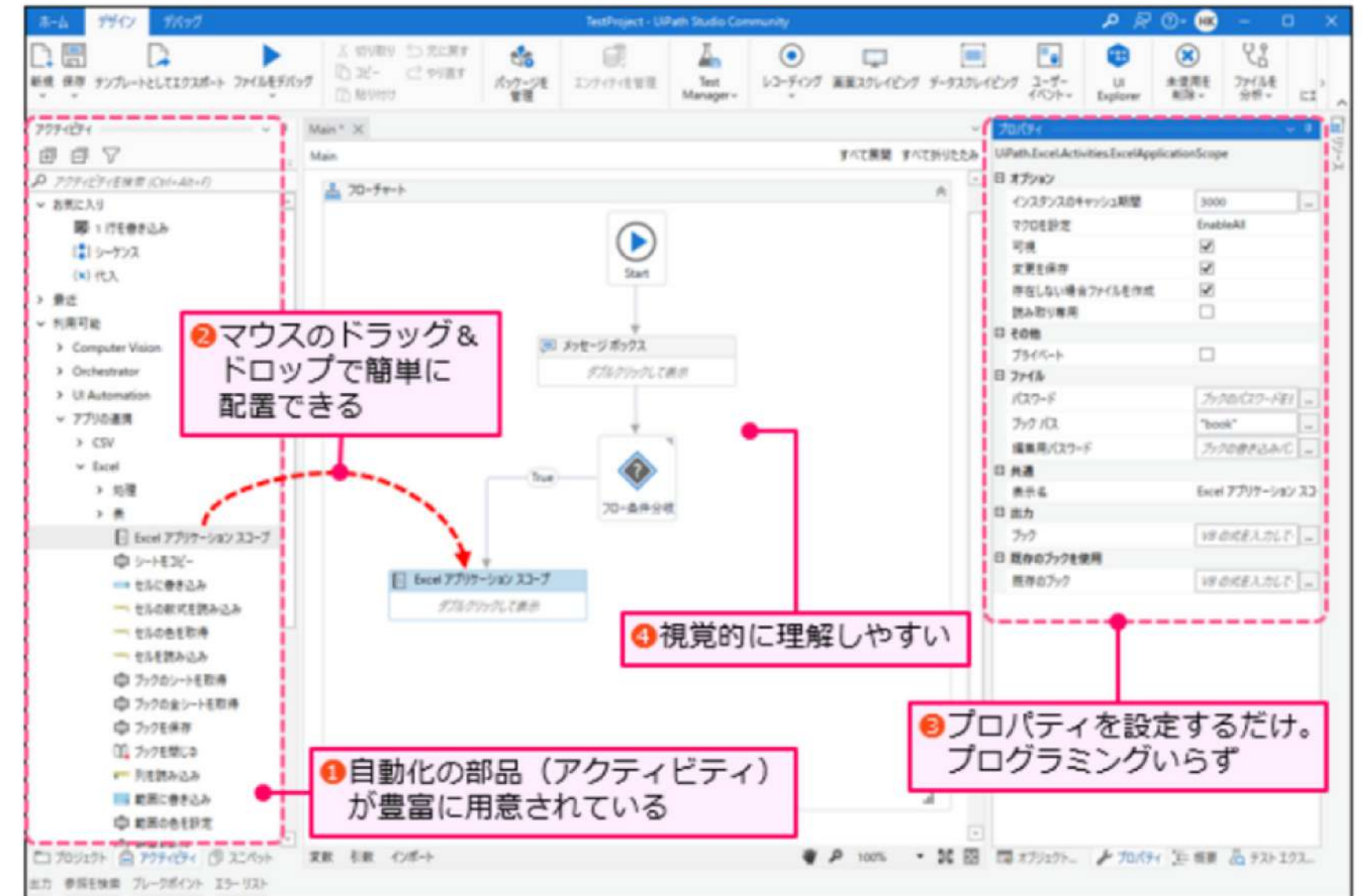
POINT!



プログラミングとは、パソコンの中のロボットにどうやってそれをすればよいか教えること。

小さな子どもに教えるように、このくらい分かるだろうということまでひとつひとつ懇切丁寧に教えてあげよう。

UiPath操作画面イメージ



参考画像出典：(株)完全自動化研究所ブログ「UiPathとは」
https://marukentokyo.jp/uiopath_feature/

※ UiPath（RPAツール）を使って
プログラミングの基礎を学習しよう！

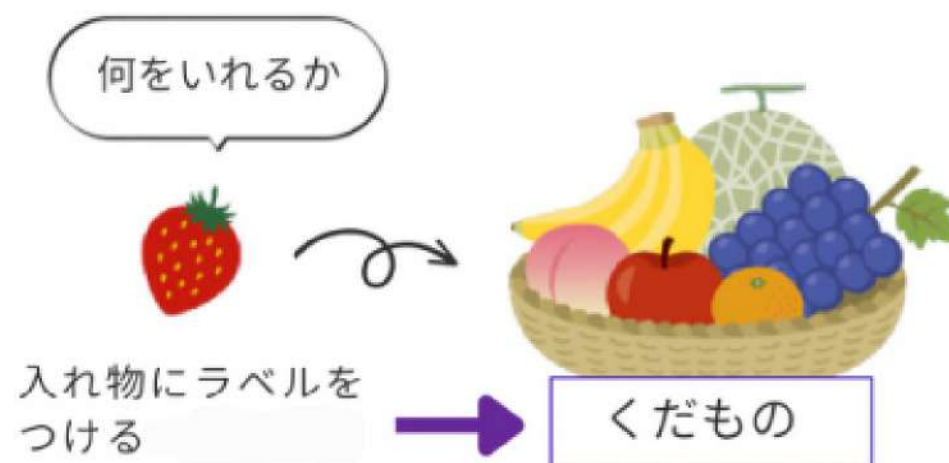
※ UiPath株式会社が提供するRPAツール



❗ ラベリング収納がプログラムのはじめの一步

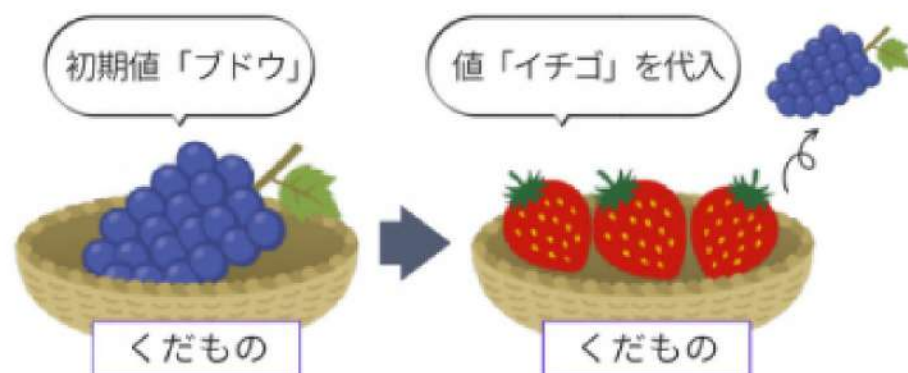
入れ物に名前を付けるのがプログラムの基本。何を入れるかによって入れ物は変わる。

代表的な入れ物に変数・定数がある。変数(定数)は一つの入れ物に一つのデータしか保存できない。



変数

変数「くだもの」の中にあるブドウをイチゴに入れ替える。変数は中身を何度でも入れ替えることができる。



定数

定数は最初に入れた値(初期値)をずっと使う。(プログラム上初期値が変わらないものが定数)
※定数も変数と呼ぶRPAが多い。



データ型を意識する

データ型とはデータ（値）の種類のこと。プログラムは文字は「文字」、数字は「数字」と、データの種類を厳密に区別している。

主なデータ型

型名	値	説明
文字列型	計算できない文字	変数は文字列型になる
数値型	数字	計算できる数字
ブーリアン型 (論理型)	true、false	「はい」か「いいえ」

これ、なんと読みますか？



左の808は計算できる数字？
できない数字？

「やおや＝文字列型」なのか「はっぴゃくはち＝数値型」なのか、データ型を教えて認識できるようにする必要がある。



❗ プログラミングの基本

上から順番に処理

プログラム順番が大事。ほぼ書いた順(上から順番)に処理される。(例外として条件分岐や繰り返しがある。)
たまに人が行う行動とは違う順番に作る必要があるので注意しよう。

繰り返し(ループ)

繰り返しの基本

- ① 「繰り返しの条件」に当てはまったら処理する
- ② 当てはまらなかったら繰り返しから抜ける。あるいは、処理するものがなくなったら終わる。

繰り返しを抜ける条件をきちんと指定し、無限ループにならないようにしよう。

アクティビティ (UiPathが用意しているプログラムのまとめ)

Excel ファイルを使用

種類: ExcelApplicationCard
名前: Excel ファイルを使用
オートメーションで使用する Excel ファイルを開く、または作成します。(Use Excel File)
UiPath.Excel.Activities に含まれています。

Excel

☐ 変更を保存 ☐ 存在しない場合ファイルを作成

読み込む値の書式: プロジェクトと同じ

☒ テンプレートファイル

{ } *01リネーム前#0909さくら.xlsx

実行

(x) 代入

保存先: { } ファイル名

保存する値: { } CurrentFile.Name

(x) 代入

保存先: { } 最終更新日

保存する値: { } CurrentFile.LastModifi

セルの値を読み込み

セル: { } Excel.Sheet("表紙").Cell("C7")

保存先: { } 担当者

☐ 書式付きテキストを取得

上のプログラムから順番に処理される



条件分岐

条件分岐はある条件を満たした時だけ実行されるプログラム。

「もし〇〇なら△△。そうでなければ□□」

日常的に人が行っている判断と同じ。

例) もし定食が1000円以下なら ➡ 食べられる
定食が1000円を超えたら ➡ 食べられない



プログラムはどのように判定するか？

条件分岐をするために、まず条件に合うかどうかを判定する必要がある。そこで演算子を使ってtrue/falseを判定させる。

条件	判定
条件に合う(成り立つ)	true (真) ○
条件に合わない(成り立たない)	false (偽) ✕

プログラミングは人の考えや行動をそのままなぞるだけ。
場面に合わせて、一番合うプログラムを選んでいこう。

条件分岐 (else if)

条件

{ } 提出日 < 提出期日

Then

{ } 本文

(x) 代入

保存先

{ } ファイル名

保存する値

{ } "A"+支店コード+"["+都

Else

{ } 本文

(x) 代入

保存先

{ } ファイル名

保存する値

{ } "B"+支店コード+"["+都

(x) 代入

保存先

{ } 提出期日

保存する値

{ } 2023-09-05

条件分岐のアクティビティ
提出日が

- 提出期限内だったらA
- 遅れたらB



① プログラミングのための計算（演算子）

四則演算

記号	説明
+	足し算（加算）
-	引き算（減算）
*	かけ算（乗算）
/	割り算（除算）
mod	あまり（剰余）

「あまり」は日常生活では使われないがプログラミングの世界では重要。例えば、偶数かどうかを調べるとき、2で割ってあまりが0なら偶数、のように使う。

比較演算子

記号	説明
A=B	AとBは等しい
A<B	AはBより小さい
A<=B	AはB以下
A>B	AはBより大きい
A>=B	AはB以上
A<>B	AとBは異なる

A=Bの場合は処理を続ける、AとBが同じでなくなったら処理をやめる、といった指示を出す際に使う。繰り返し（ループ）で使うことが多い。プログラミングごとの記号の書き方があるので注意。

論理演算子

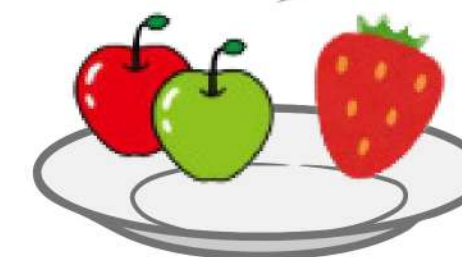
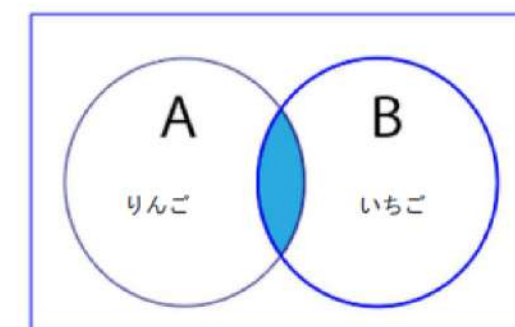
論理演算子	説明
AND	A かつ B AとB両方trueの時⇒true
OR	A または B AかBどちらかtrueの時⇒true
NOT	Aがtrueでないとき⇒true

★「AND」を使った例

お皿にリンゴとイチゴがのっていたら
食べられる。



食べられる！



RPAシナリオを作る上で必須！ プログラムの基本まとめ

✓ ラベリング収納がプログラムのはじめの一步

- 最初に理解したい 変数、定数
- データ型も意識しておこう

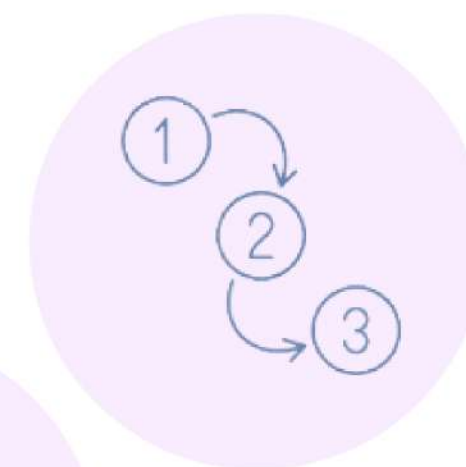
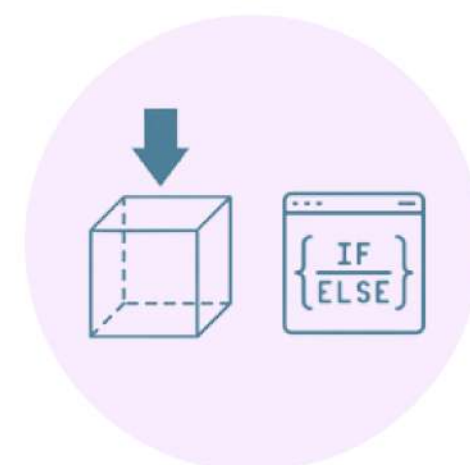
✓ プログラミング

- 上から順番に処理
- 繰り返し(ループ)
- 条件分岐

✓ プログラミングのための計算

- 四則演算 ・ 比較演算子
- 論理演算子

Excel関数、PowerQuery、マクロ、Python…その土台はプログラム！
プログラミングの基礎知識を習得し
ITスキルアップにつなげましょう！



全4回のExcel講習
お疲れ様でした

