

WinActor情報局
Live

NTT Data
Trusted Global Innovator

第2回：すぐできる！シナリオ作成のコツ

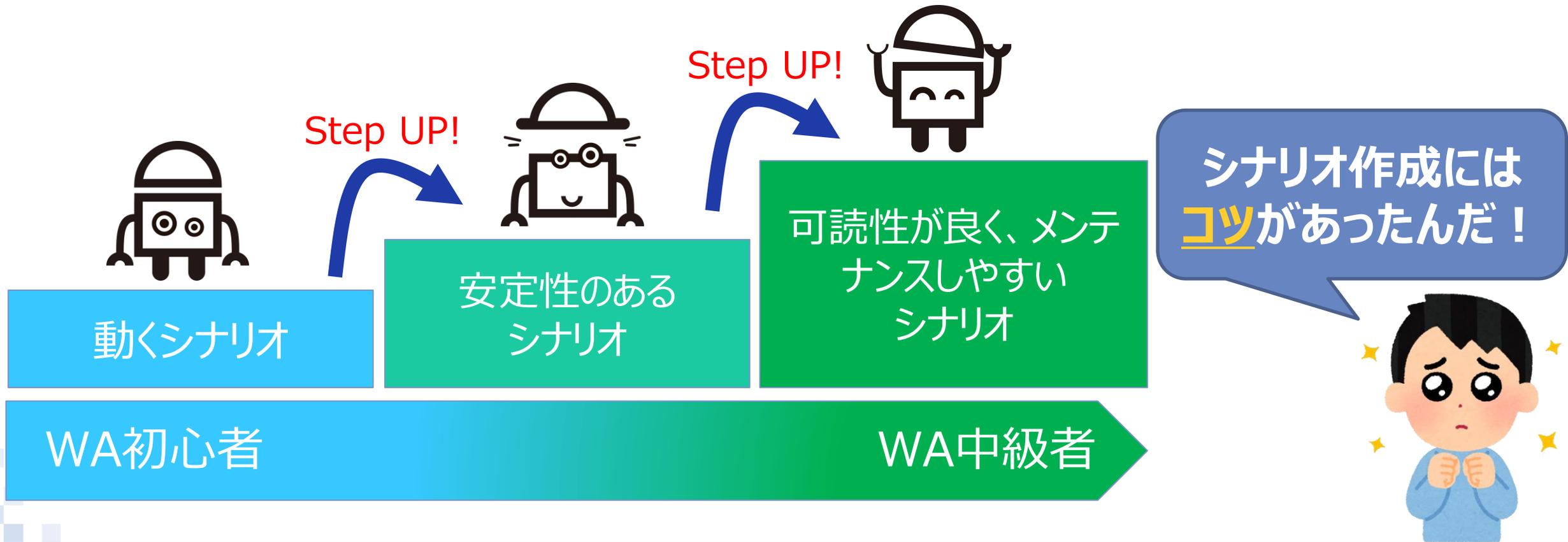
NTT DATA RPAソリューション担当

本日の流れ

1. はじめに
2. すぐできる！シナリオ作成のコツの紹介
 - コツ①シナリオ作成の心得
 - コツ②シチュエーションに合わせたライブラリ設定
 - コツ③シナリオの簡略化
 - コツ④安定したシナリオ作成
 - コツ⑤メンテしやすいシナリオの作成方法
 - コツ⑥エラー処理の実装と回避
 - 必殺！小技集
3. まとめ

1. はじめに

ステップアップするには、シナリオ作成時に意識すべきコツがあります。
そのコツをつかんでWA技術者としてステップアップしていきましょう！！

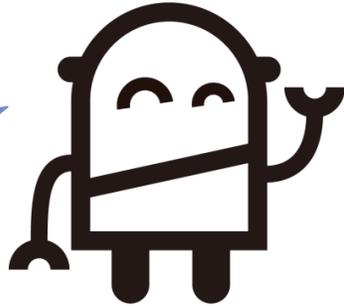


2. コツ①シナリオ作成の心得

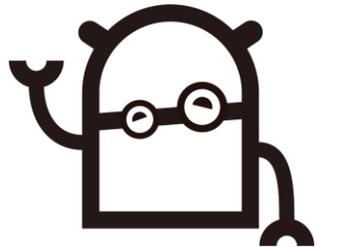


シナリオ作成の心得とは？

フローや変数一覧を見ただけで処理内容を見やすくなるように工夫することだよ



可読性がUPするとメンテナンスもしやすいよね！



2. コツ①シナリオ作成の心得

■ 変数一覧の整理、変数名の付け方

① グループ名ルール

【対象】という形式で命名する。

例：「データ一覧」というグループ名を作成し、データ一覧ファイル操作で利用する変数を登録する。

② 変数名ルール

【対象】_【内容】という形式で命名する。

例：「管理簿」というファイルに対して「最終行」という変数を作成する場合、「管理簿_最終行」となる。

※「対象」、「内容」の定義が冗長にならないように気をつけること。

2. コツ①シナリオ作成の心得

■ 変数一覧の整理、変数名の付け方

例) 「データ一覧」、「管理簿」ファイルに対して操作を行う。

<Before>

グループ名	変数名
グループなし	実行日付
	個人ID
	結果
	シート名
	ファイル名
	ファイルパス1
	行
	最終行
	ファイルパス

処理対象ファイルが2つあるのに、
どのファイルに対する変数なのか
分からないなあ・・・



2. コツ①シナリオ作成の心得

■ 変数一覧の整理、変数名の付け方

<After>

グループ名	変数名
▼ 共通	実行日付 個人ID 結果
ルール①	
▼ データ一覧	データ一覧_ファイルパス データ一覧_シート名 データ一覧_ファイル名
▼ 管理簿	
ルール②	管理簿_行 管理簿_最終行 管理簿_ファイルパス
▼ グループなし	
▼ 新しいグループ	

グループ化がされていて見やすい！
どのファイルに対しての変数なのかが
一目で分かる！



2. コツ①シナリオ作成の心得

■ 使用ノードをグループ化し名前をつける

① グループ名ルール

【対象】+【動作】という形式で命名する。（処理内容を示す）

※名前欄だけでは、グループ操作・目的が判断できない場合はコメント欄に記載すること

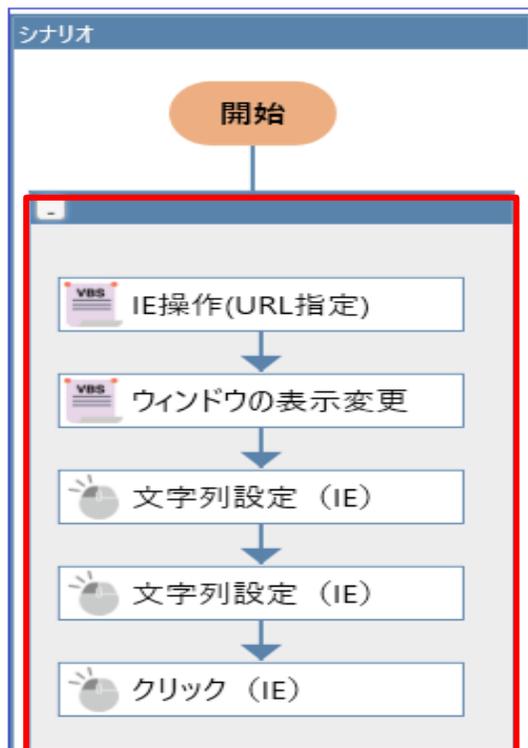
例：顧客IDを入力する/顧客ID入力

2. コツ①シナリオ作成の心得

■ 使用ノードをグループ化し名前をつける

<Before>

例：基幹システムを開き、ログインする処理



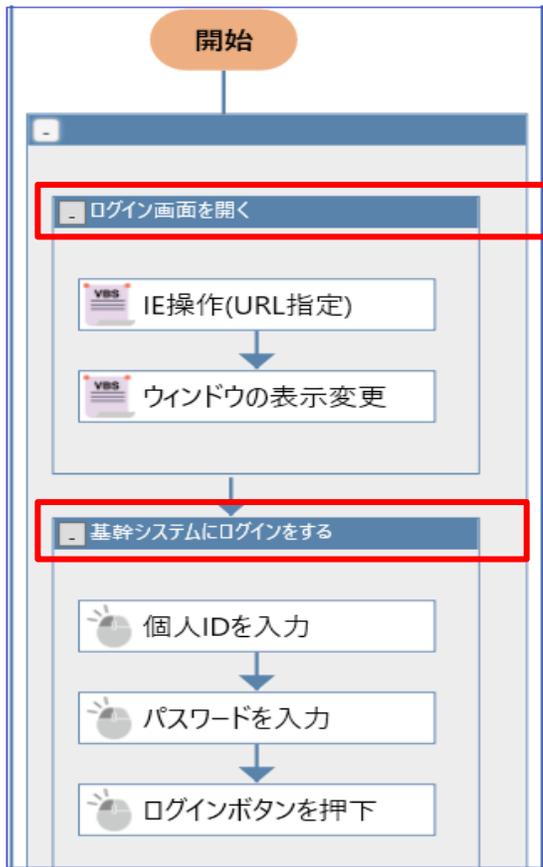
グループ化されておらず、ノードにも名前が付いていないため、どの処理に対して何をしているのか分からない...



2. コツ①シナリオ作成の心得

■ 使用ノードをグループ化し名前をつける

<After>



グループ名が処理内容になっていて見やすいし、分かりやすい！
これで誰が見ても分かるシナリオになった！

グループ名を順番に確認すると、処理の全体像が把握出来る！

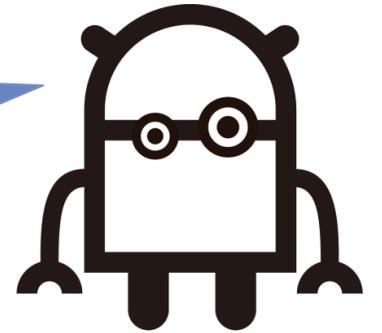


2. コツ②シチュエーションに合わせたライブラリ設定

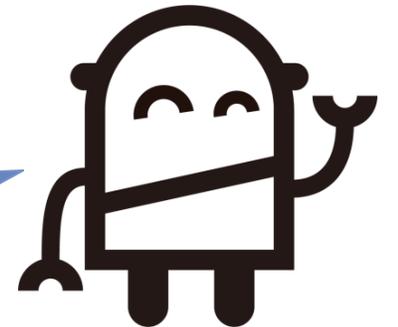


いつも同じようなライブラリしか使わないな

ノードやライブラリの用途を理解できていないからなんじゃないかな？



条件に適したライブラリを選択することができる！



2. コツ②シチュエーションに合わせたライブラリ設定

■ Excel処理の繰り返しパターンの紹介

Excel上の登録データが可変の場合、要件によって実装方法を使い分ける必要がある！

No	要件	実装方法	備考
1	空白データまで繰り返す必要がある。	「分岐」を使用し繰り返す方法	空白以降にデータが登録されていても、繰り返しは終了する。
2	空白データ以降もデータが登録されている場合、繰り返す必要がある。	「Excel操作(最終行取得その4)」を使用し繰り返す方法	空白データはスキップする。

2. コツ②シチュエーションに合わせたライブラリ設定

■ CSVファイルの読み込みパターンの紹介

CSVファイルを読み込む場合、まずは、読み込む要件を確認する!

	A	B	C	D	E	F	G	H
1	06_調査時期	06_地域	06_総人口【人】	06_総人口(06_総人口(06_65歳以	06_総人口(06_65歳以	06_65歳以	06_65歳以	06_65歳以上
2	20190515240000	埼玉県	7,350,000	3,668,000	3,682,000	26.7	24.1	29.2
3	20190515240000	千葉県	6,259,000	3,105,000	3,155,000	27.9	25.3	30.3
4	20190515240000	東京都	13,921,000	6,846,000	7,075,000	23.1	20.3	25.7
5	20190515240000	神奈川県	9,198,000	4,585,000	4,613,000	25.3	22.6	28
6								
7								
8								
9								
10								
11								
12								

必要な項目（列）を読み込む？

1行全てを読み込む？

今後の変更を考えると列の入れ替えもありそう…
どのライブラリを使うのがベストかな？



2. コツ②シチュエーションに合わせたライブラリ設定

■ CSVファイルの読込パターンの紹介

No	要件	利用ライブラリ	補足
1	取得したい列の順番が変わらない	Excel操作 (行の読み取り)	取得項目数が増えるとスクリプト修正が発生する。
2	列の順番が可変になる	CSVファイル→変数値 (※1)	事前に変数名をインポートする必要がある。

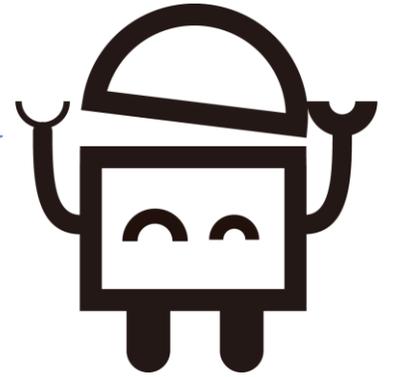
(※1) CSV上に項目情報が付与されていない場合は、利用出来ない。

2. コツ③シナリオの簡略化

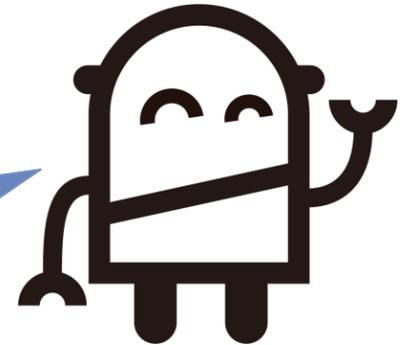


画像マッチングやエミュレーションで対応してはダメなの？

プチライブラリを使うとフローがスッキリするよ



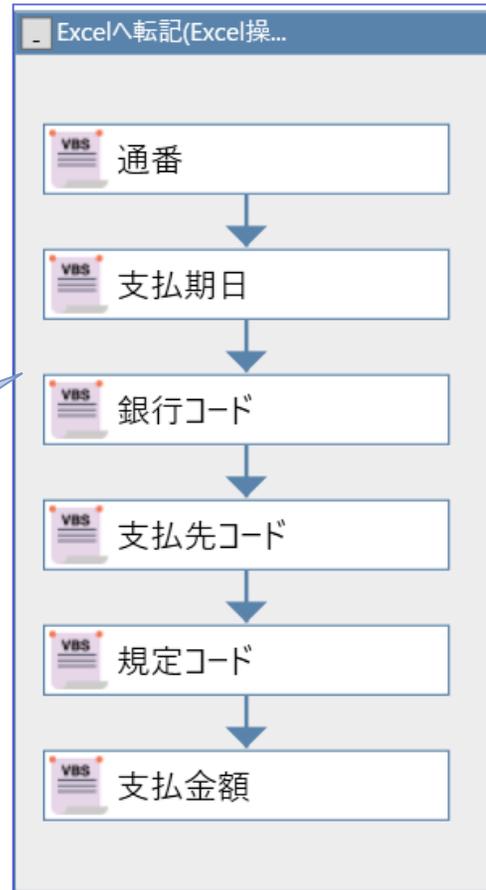
使用ノードを減らすことで、シナリオ作成時間も削減できて、見やすいシナリオになる！



2. コツ③シナリオの簡略化

プチライブラリ「Excel操作(行の書き込み)」の使用 基本ライブラリ「Excel操作(値の設定)」で実装すると…

設定したい列分だけライブラリを使用しなければならない。

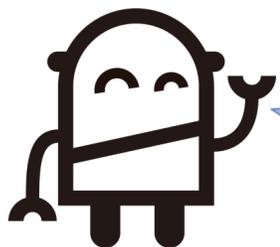


書き込みたい項目がたくさんあったら大変だ～



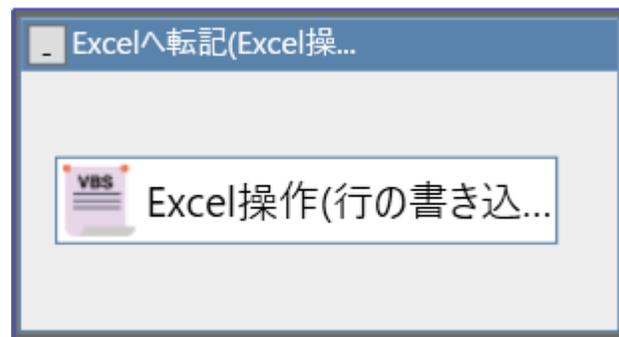
2. コツ③シナリオの簡略化

プチライブラリ「Excel操作(行の書き込み)」の使用



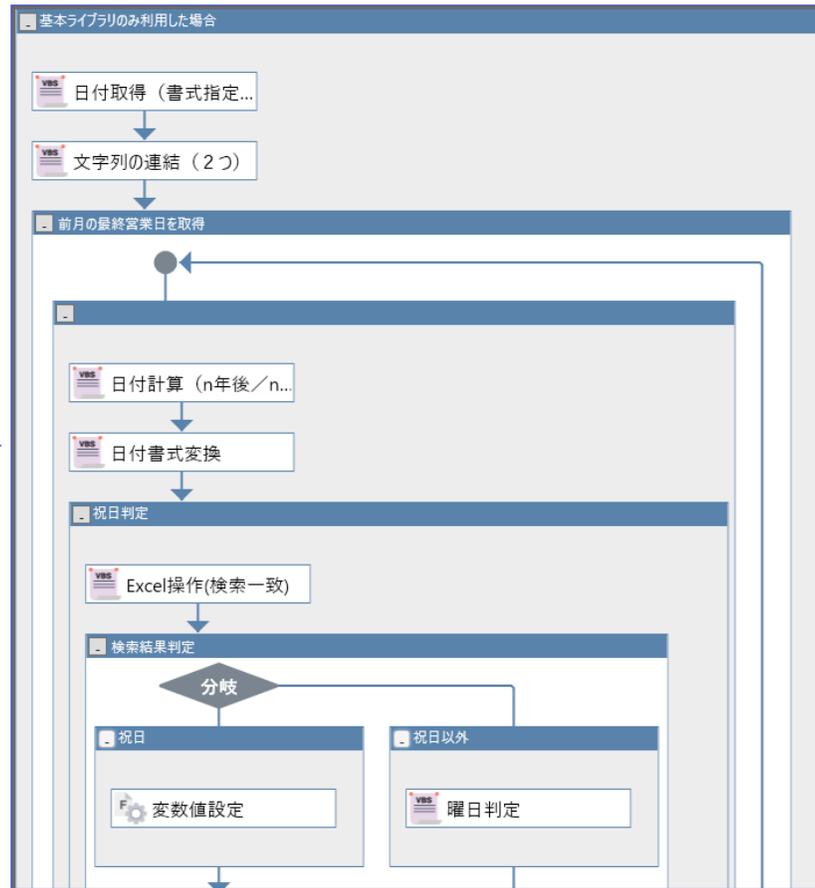
プチライブラリ「Excel操作(行の書き込み)」で実装してみてください！

こんなに**簡略化**できるのね
1つのライブラリで実装できるなんて感動！



2. コツ③シナリオの簡略化

■ 前月の最終営業日を取得する方法 基本ライブラリで実装すると・・・



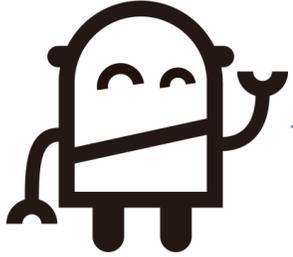
処理が複雑になり、可読性も下がる

解読するのも大変だ・・・
こんな複雑な処理作れる気がしないよ～

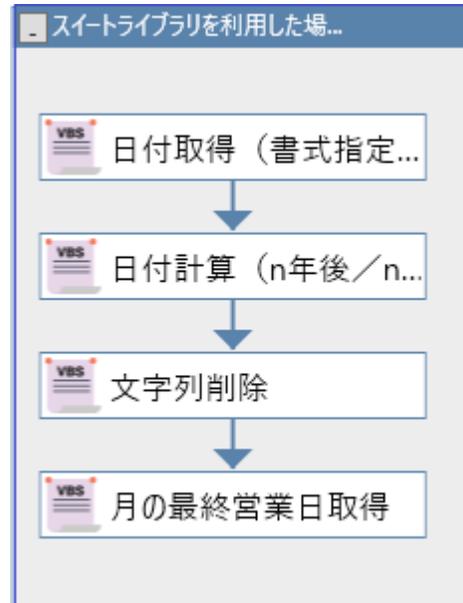


2. コツ③シナリオの簡略化

■ 前月の最終営業日を取得する方法



スイートライブラリ「月の最終営業日取得」で実装してみてください！

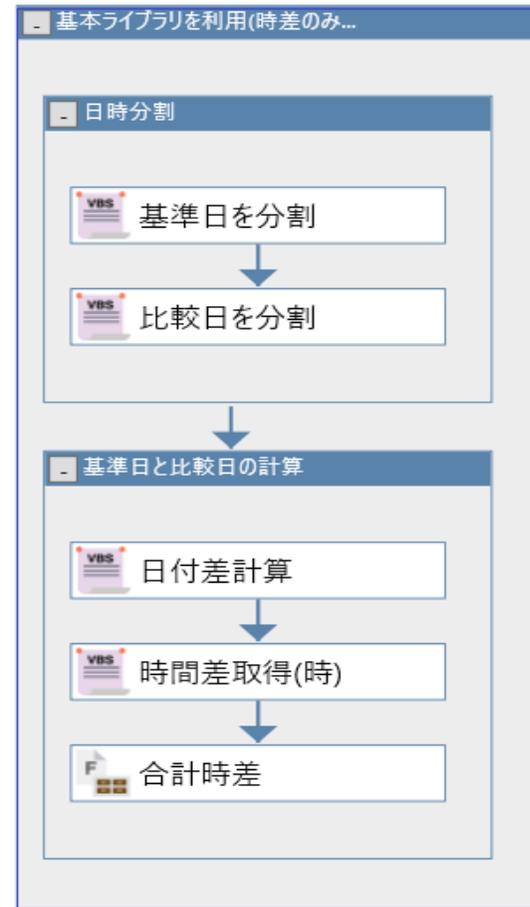


処理内容が分かるようになった！
スイートライブラリは使ったことなかったけど、他にも使えそうなものがありそう！



2. コツ③シナリオの簡略化

■ 2つの日付の時間差の算出方法 基本ライブラリで実装すると・・・

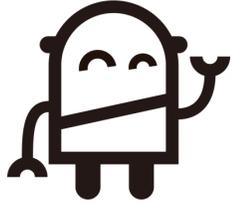


1つのライブラリでできる方法はないのかな

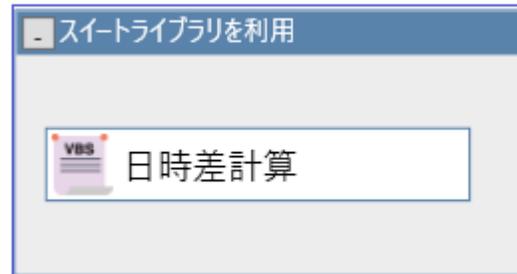


2. コツ③シナリオの簡略化

■ 2つの日付の時間差の算出方法



スイートライブラリ「日時差計算」で実装してみてください！



求めてたものが**スイートライブラリ**にあった！

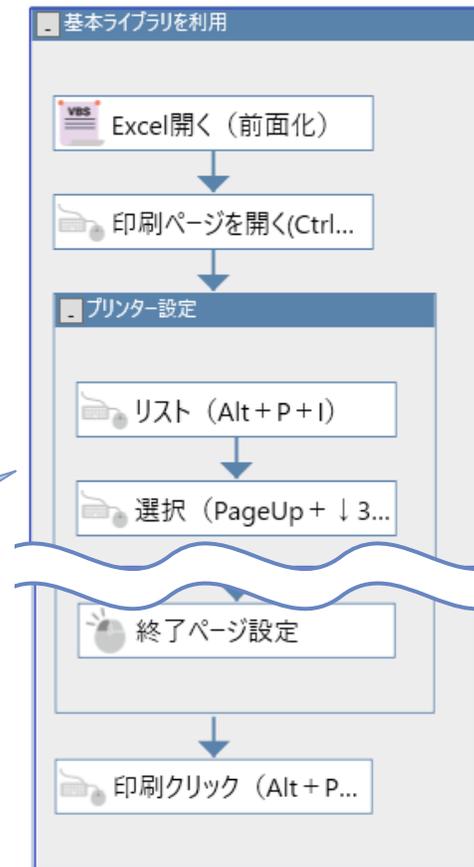


2. コツ③シナリオの簡略化

■ 印刷方法について

基本ライブラリで実装すると・・・

Excelのショートカットキーをエミュレーションで実装してる。

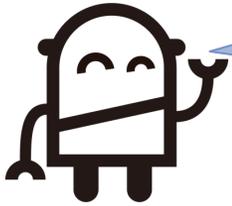


Excelのショートカットキーをエミュレーションで実装するから、ノード数が多くなるし、安定性も心配だな・・・

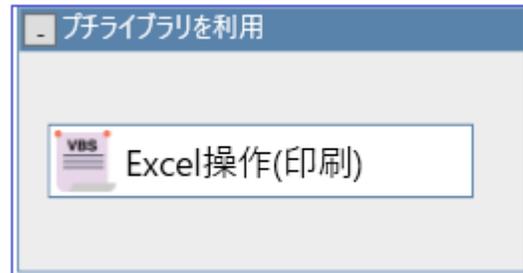


2. コツ③シナリオの簡略化

■ 印刷方法について



プチライブラリ「Excel操作(印刷)」で実装してみてください！



ノードも数も削減できるし、ライブラリを使用するから、これなら**安定性もUP**する！



ひと休み♪



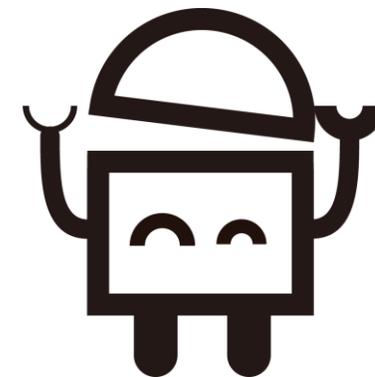
2. コツ④安定したシナリオ作成



シナリオの動作が安定しないな・・・

ライブラリをちゃんと使用してるかな！

適切なライブラリを使用するだけで
エラーの発生を減らせるよ！

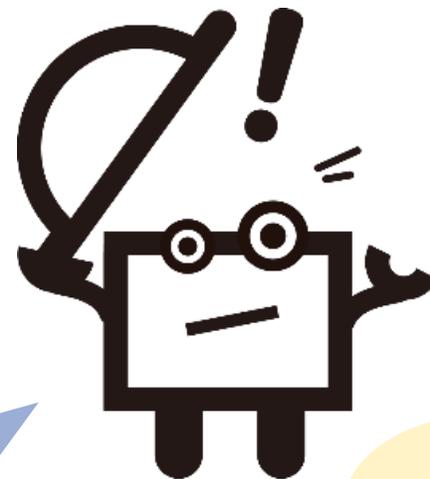


2. コツ④安定したシナリオ作成

■ライブラリの活用

よりシナリオの動作を安定させるには・・・？

エミュレーションモードや画像マッチングで
たま～にエラーが出る箇所があるなあ



✓CHECK

複雑な処理は
適切なライブラリを使うことで
安定化させよう！

⇒【NEXT】Excelのデータをソートするシナリオ

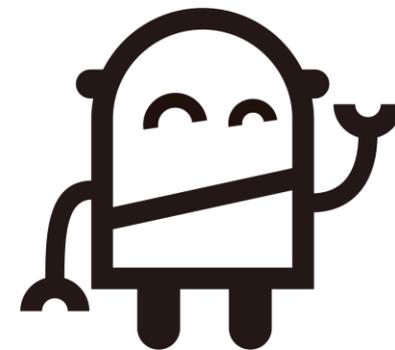
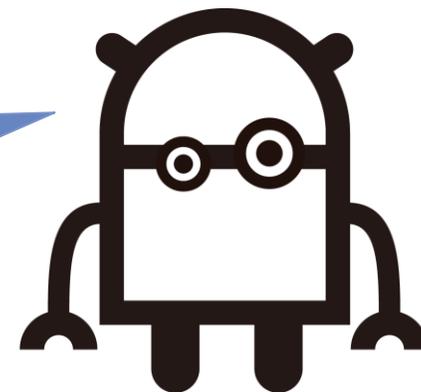
2. コツ⑤メンテしやすいシナリオ作成方法



シナリオのメンテナンスがうまくいかない・・・

誰が見ても理解ができるシナリオになっているかな？

処理の共通化と可読性を意識して作ってみよう！

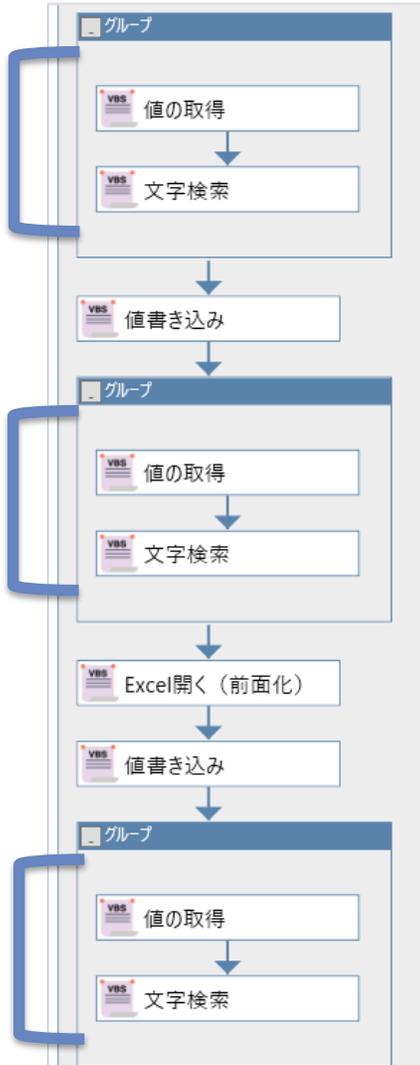


2. コツ⑤メンテしやすいシナリオ作成方法

■ サブルーチンの活用

共通処理が複数存在すると...

同内容の処理が
シナリオに複数回
配置される



⇒【NEXT】Excelに値を記入し、保存するシナリオをサブルーチン化

2. コツ⑤メンテしやすいシナリオ作成方法

■ファイルパスの設定方法

The image shows two screenshots from the WinActor software. The top screenshot is the '変数一覧' (Variable List) window, which has a table with columns: グループ名 (Group Name), 変数名 (Variable Name), 現在値 (Current Value), 初期化しない (Do not initialize), 初期値 (Initial Value), and マスク (Mask). A red box highlights the row where the variable name is 'ファイルパス' (File Path) and the initial value is 'C:\Users\%xxxxxxx\Desktop\作業ファイル.xlsx'. The bottom screenshot is the 'プロパティ' (Properties) dialog for a script named 'Excel開く (前面化)'. It has tabs for '設定' (Settings), 'スクリプト' (Script), '注釈' (Notes), and 'バージョン情報' (Version Information). The '設定' tab is active, showing instructions for file paths and a dropdown menu for 'ファイル名' (File Name) with 'ファイルパス' selected. A red box highlights this dropdown.

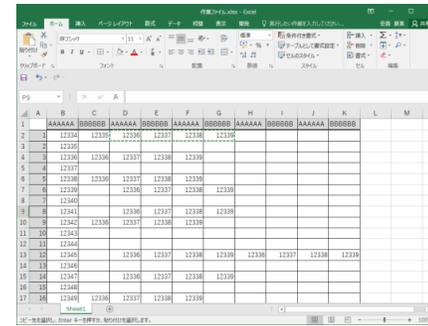
変数一覧

グループ名	変数名	現在値	初期化しない	初期値	マスク
グループなし	ファイルパス		<input type="checkbox"/>	C:\Users\%xxxxxxx\Desktop\作業ファイル.xlsx	<input type="checkbox"/>

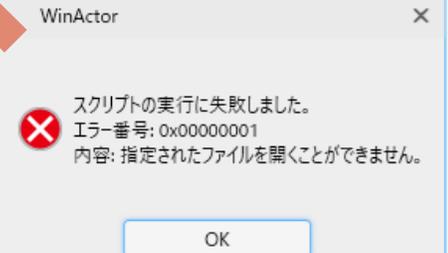
ファイルパスを
Aさんのローカルフォルダ
に設定

実行
すると...

Aさんの端末



Bさんの端末



2. コツ⑤メンテしやすいシナリオ作成方法

■ファイルパスの設定方法

1 ライブラリ「シナリオフォルダのファイルパス」を使用
→シナリオフォルダを起点としたファイルパスを作成することができる

プロパティ

スクリプト実行

名前 シナリオフォルダのファイルパス

コメント

設定 スクリプト 注釈 バージョン情報

シナリオフォルダを起点としたファイルパスを作成します
シナリオが保存されていない場合は、空の情報となります。

例) シナリオが c:\temp フォルダに保存されており、
ファイル名に a.txt を指定した場合、
ファイルパス生成結果には c:\temp#a.txt が格納されます。

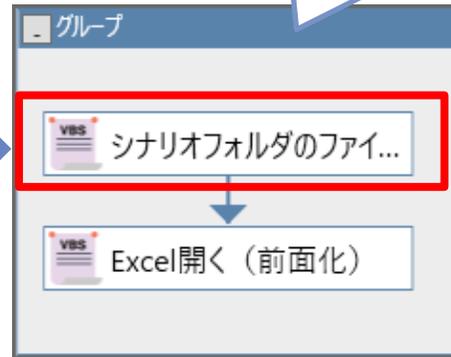
ファイル名 値⇒ 作業ファイル.xlsx

ファイルパス生成結果 作業ファイル_ファイルパス

更新 元に戻す

設定

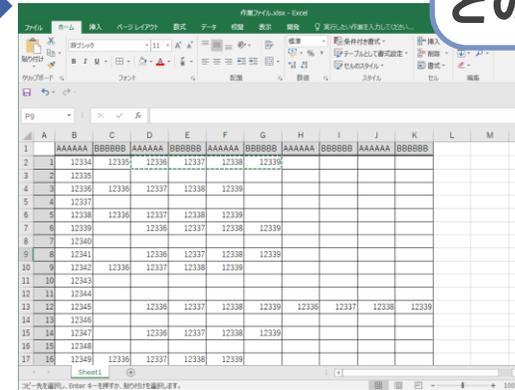
シナリオフォルダを起点と
したファイルパスを生成



実行



シナリオフォルダ内の
ファイルであれば、
どの端末でも開くことが可能!



2. コツ⑤メンテしやすいシナリオ作成方法

■ファイルパスの設定方法

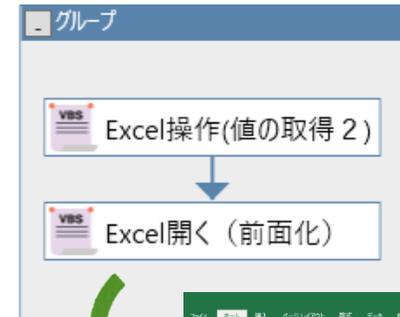


外部ファイルでファイルパスを管理し、シナリオに読み込ませる

1.ファイルパスを記載した管理用外部ファイルを作成▼

項番	ファイル名	ファイルパス
1	作業ファイル.xlsx	¥¥xxxx-xx¥RPA¥管理¥01¥02_RPA作業¥作業ファイル¥作業ファイル.xlsx
2	月次_集計用ファイル.xlsx	¥¥xxxx-xx¥RPA¥管理¥02¥02_集計作業¥6月¥月次_集計用ファイル.xlsx
3	名簿.xlsx	¥¥xxxx-xx¥RPA¥管理¥04¥03_管理名簿¥ウサギさんチーム¥名簿.xlsx

2.「Excel操作(値の取得2)」等で外部ファイルからファイルパスを読み取り、対象ファイルを開く



ファイルパスを変更する場合、管理用ファイルを更新するだけ！



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	AAAAA	BBBBB											
2	12334	12335	12336	12337	12338	12339							
3	12335												
4	12336	12336	12337	12338	12339								
5	12337												
6	12338	12336	12337	12338	12339								
7	12339												
8	12340												
9	12341	12336	12337	12338	12339								
10	12342	12336	12337	12338	12339								
11	12343												
12	12344												
13	12345	12336	12337	12338	12339	12336	12337	12338	12339				
14	12346												
15	12347	12336	12337	12338	12339								
16	12348												
17	12349	12336	12337	12338	12339								



2. コツ⑥エラー処理の実装と回避



今までエラーが起きていなかったところでエラーが発生してしまった！

エラー原因の**特定**や**解析**はできているかな？



想定外のエラー発生に対応したシナリオ作成をしよう！



2. コツ⑥エラー処理の実装と回避

■エラー時の対応

想定外のエラーが発生した時に…

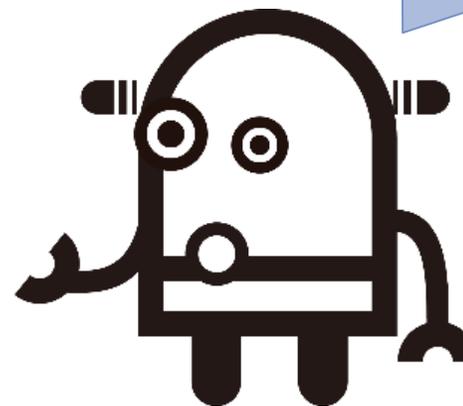
エラーの原因が分からない



エラー情報をメモし忘れた

✓CHECK

エラー発生時の対応は、
あらかじめシナリオに組み込むことが
できるよ！

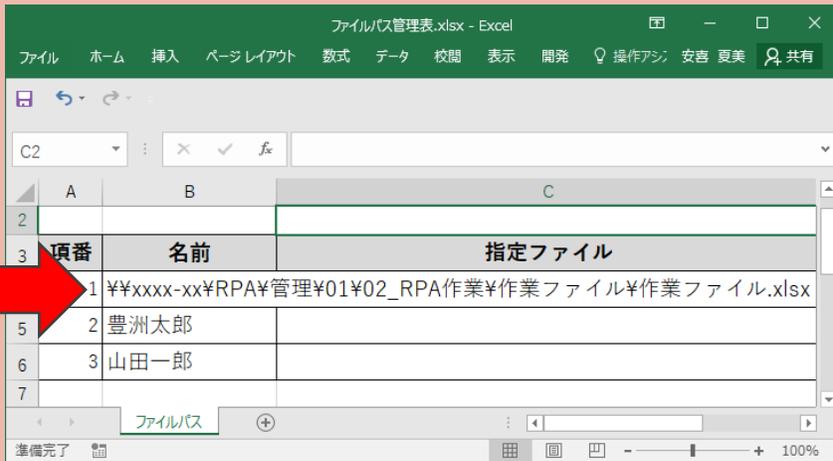


2. コツ⑥エラー処理の実装と回避

■エミュレーションモードの留意点

エミュレーションが想定通りの挙動をせず、困ってしまった・・・

▼正しい箇所に文字入力ができない



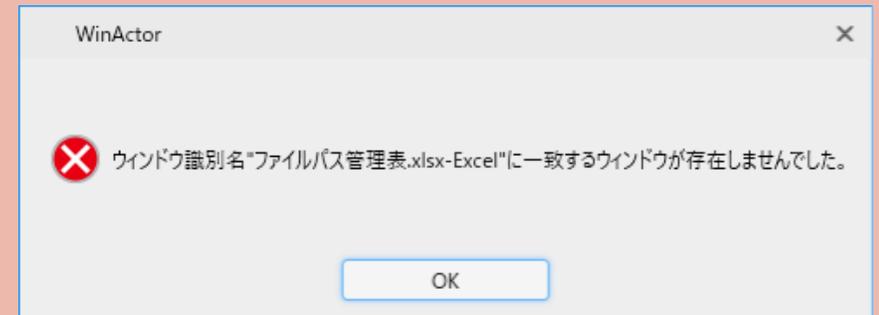
The screenshot shows an Excel spreadsheet titled "ファイルパス管理表.xlsx - Excel". The spreadsheet has three columns: "項番" (Item No.), "名前" (Name), and "指定ファイル" (Specified File). The data rows are:

項番	名前	指定ファイル
1	¥¥xxxx-xx¥RPA¥管理¥01¥02_RPA作業¥作業ファイル¥作業ファイル.xlsx	
2	豊洲太郎	
3	山田一郎	

A red arrow points to cell C2, which is currently empty. The status bar at the bottom indicates "準備完了" (Ready) and "100%" zoom.



▼操作対象が判定されない



2. コツ⑥エラー処理の実装と回避

■エミュレーションモードの留意点



エミュレーションモード使用時に、エラーを回避するための留意点

留意点	内容
①マウス操作の記録	シナリオ作成時と実行時に、操作対象が同じ位置になるよう調整する。
②待機時間の設定	一つの操作ごとに適当な待機時間を設定する。

2. コツ⑥エラー処理の実装と回避

■エミュレーションモードの留意点

①マウス操作の記録

対象をクリックするシナリオ

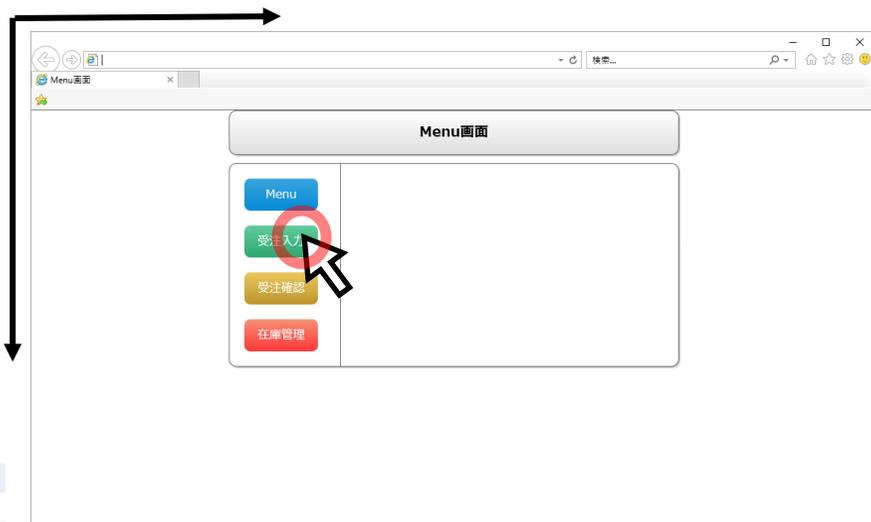
シナリオ作成時の座標位置

原点: 左上
X座標: 327
Y座標: 413

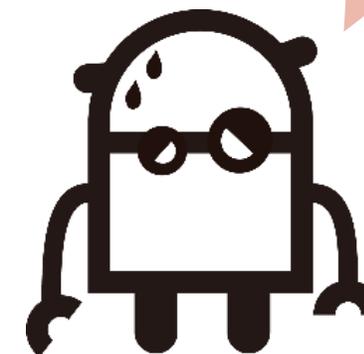


シナリオ実行時の座標位置

原点: 左上
X座標: 327
Y座標: 413



操作対象の位置が
違うから、想定通りの箇所
をクリックされないね



2. コツ⑥エラー処理の実装と回避

■エミュレーションモードの留意点

①マウス操作の記録



1 画面の最大化

⇒ライブラリ「**ウィンドウの表示変更**」を使用し、画面を最大化してからマウス操作をする。

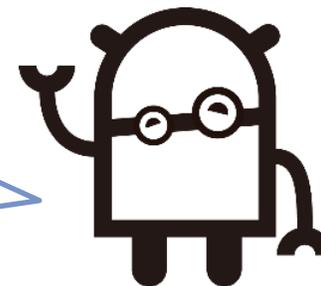


2 表示倍率の固定

⇒操作対象の画面を**特定の表示倍率に設定**してからマウス操作をする。

✓CHECK

「ブラウザ操作(ズームを100%に変更)」等の
プチライブラリもあるよ！



2. コツ⑥エラー処理の実装と回避

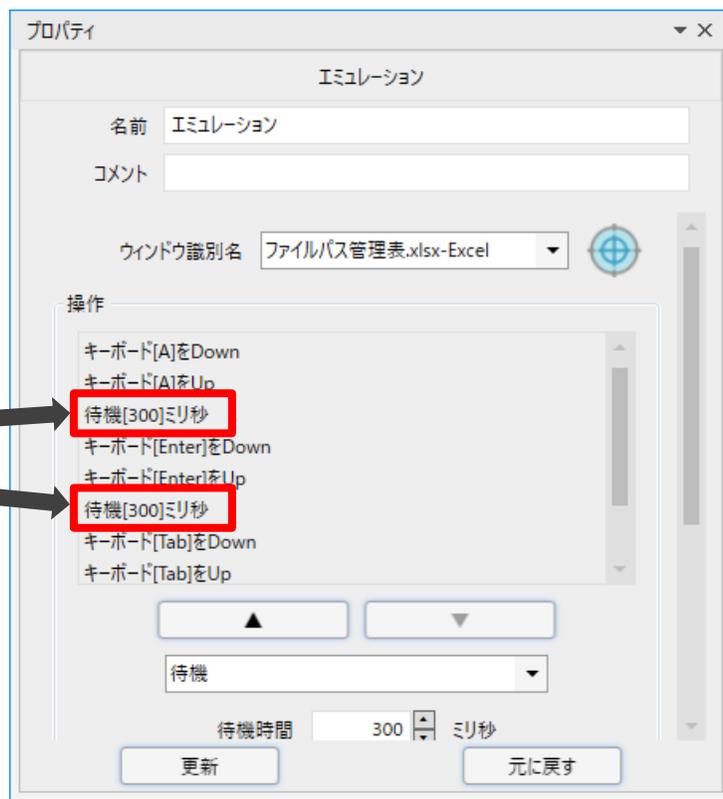
■エミュレーションモードの留意点

②待機時間の設定

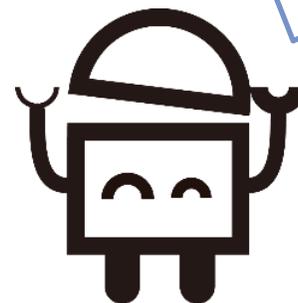


1つの操作ごとに適切な待機時間を設定する。

✓待機時間を
設定



操作対象画面が表示される前に
エミュレーションを実行してしまう
ことが防げるね！



必殺！小技集

覚えておくと便利じゃぞ

 不要なイメージ一括削除でスッキリ！

 画面を消して自由に選択！

 大量変数を一括に定義！



2. 必殺！小技集

■ 不要なイメージ一括削除でスッキリ！

シナリオ作成において、知らないうちにサイズが大きくなってしまった経験はありませんか？

未使用イメージ名を削除！！

<削除前>

名前	サイズ
サンプルシナリオ.ums7	9,117 KB

未使用イメージ名削除

未使用イメージ

削除実行

<削除後>

名前	サイズ
サンプルシナリオ.ums7	1,265 KB

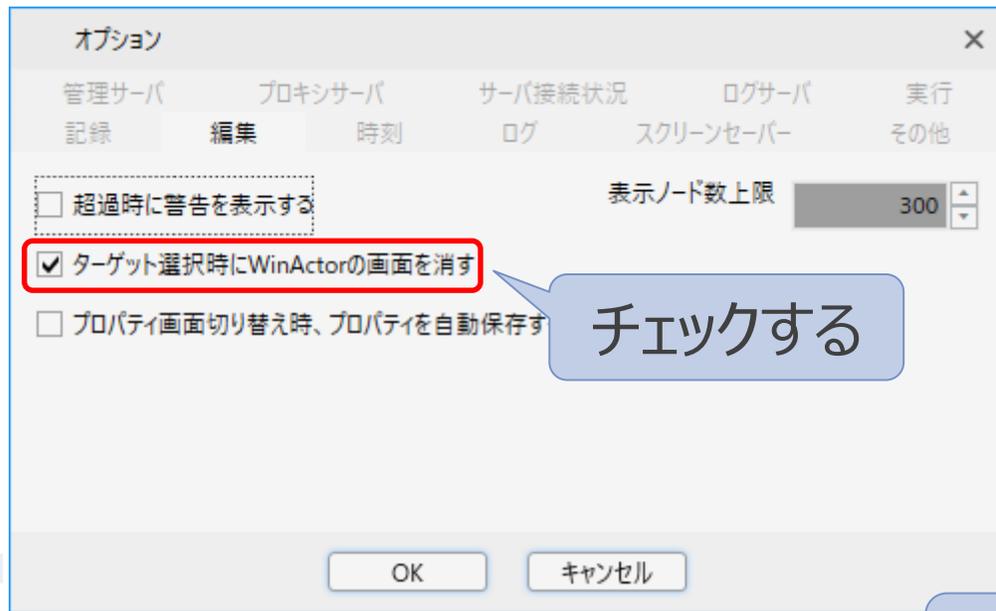
不要なイメージを削除
サイズもスッキリ！

2. 必殺！小技集

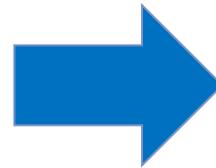
■ 画面を消して自由に選択

ターゲットを選択する際に、WinActor画面に隠れて選択しにくいと思ったことはないでしょうか？

オプション設定で解決！！



ターゲット
選択をすると



2. 必殺！小技集

■ 大量変数を一括に定義

項目数の多いcsvファイルに対する変数を手動で追加するのが面倒だと思いませんか？

変数名インポート操作で解決！！

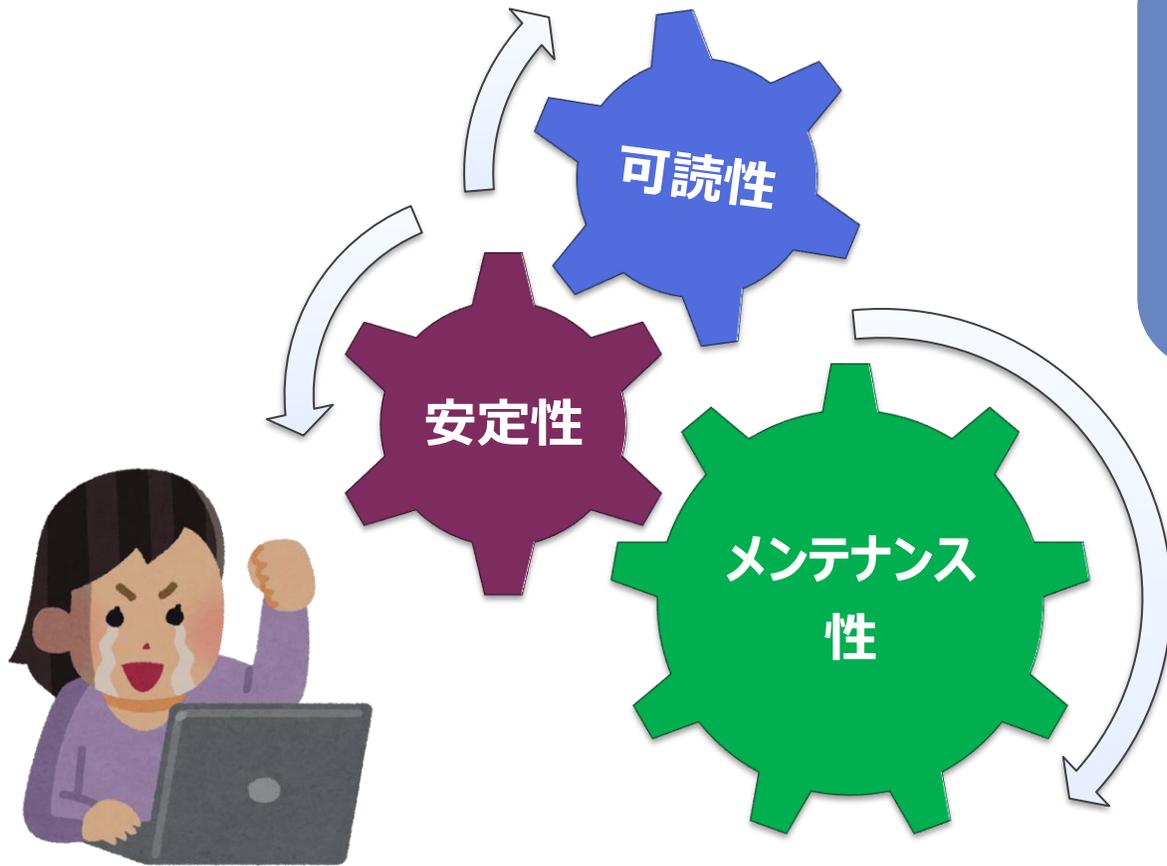
	A	B	C	D	E	F	G	H	I	J	K	L
1	項目1	項目2	項目3	項目4	項目5	項目6	項目7	項目8	項目9	項目10	項目30
2	AAAA	BBBB	CCCC	DDDD	EEEE	FFFF	GGGG	HHHH	IIII	JJJJ		ZZZZ

合計30個の変数を登録する必要がある・・・

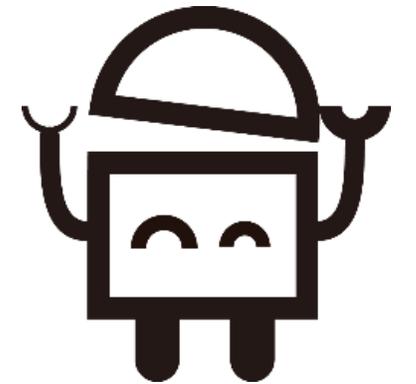


3. まとめ

各コツを意識することで、シナリオの「メンテナンス性」、「可読性」、「安定性」に繋がります。



3つ要素が噛み合うと、より良いシナリオを作成することができます。
皆さん、一緒にステップアップしていきましょう！！



3. まとめ

今まで作成したシナリオ
を見直してみるわ！

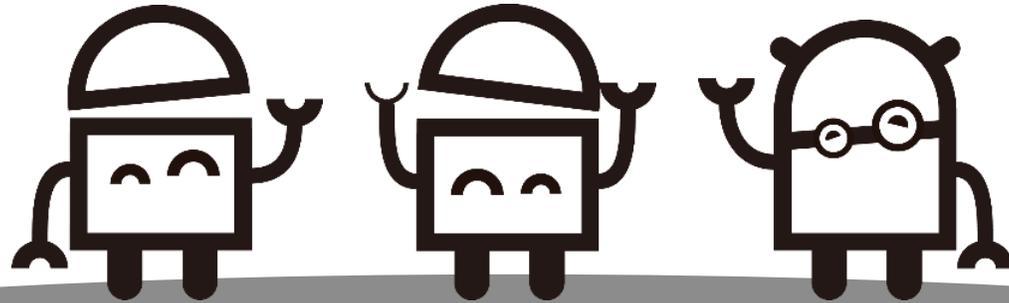
なんだかシナリオを作成
したくなってきたぞ！



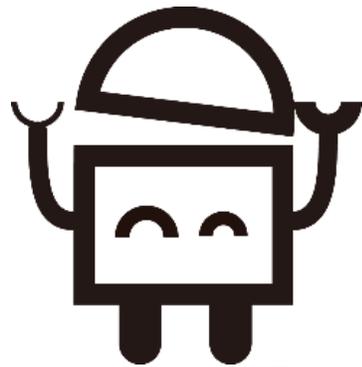
3. まとめ

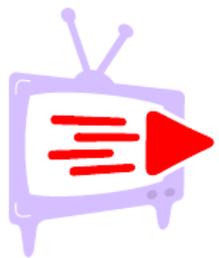
配信内容に関するお問い合わせ・アンケート送付先はこちら

■ WinActorお悩み相談窓口 : rpa-yorozuadmin@am.nttdata.co.jp



ご清聴ありがとうございました！





WinActor情報局

Live 

NTT DATA
Trusted Global Innovator