



学習シナリオマニュアル

NTT アドバンステクノロジー株式会社

目次

1. はじめに	1
2. 本書の対象読者	1
3. 学習シナリオの動作環境	1
4. 商標について	1
5. 本書に関する注意	1
6. 学習シナリオの一覧	2
7. 各学習シナリオの詳細	4
7.1. ノード、ライブラリのグループ化	4
7.2. シナリオで付箋追加	5
7.3. ブザー音発生	6
7.4. WAVE ファイル音発生	7
7.5. 簡単なメッセージ表示	8
7.6. 現在の日時取得	9
7.7. ログイン中のユーザ名取得	10
7.8. 変数値設定と表示	11
7.9. データの入力と表示	12
7.10. データの選択と表示	13
7.11. 数値の加算	14
7.12. 2つの数字の四則演算	15
7.13. 変数のコピーと表示	16
7.14. 半角から全角に変換	17
7.15. 指定時間待機後に実行	18
7.16. クリップボードで値設定・取得	19
7.17. 2つの選択肢から選択した結果表示	20
7.18. 3つ以上の選択肢から選択した結果表示	21
7.19. 操作前判定して繰り返す(条件式)	22
7.20. 操作前判定して繰り返す(回数)	24
7.21. 操作前判定して繰り返す(範囲)	26
7.22. 操作前判定して繰り返す(データ数)	28
7.23. 操作後判定して繰り返す(条件式)	30
7.24. 操作後判定して繰り返す(回数)	32
7.25. 操作後判定して繰り返す(範囲)	34
7.26. 操作後判定して繰り返す(データ数)	36

7.27.	繰り返しを途中で終了	38
7.28.	繰り返しの一部をスキップして次の条件判定に進む	39
7.29.	エラーが発生して例外処理の異常系統行	40
7.30.	別のシナリオを呼び出す	41
7.31.	呼び出し先シナリオ	42
7.32.	シナリオからサブルーチンを呼び出す	43

1. はじめに

「学習シナリオマニュアル」(以降、本書と記載)では、「学習シナリオ」(ノードを学習するため初心者向けのサンプルシナリオ)の内容、および使い方について説明します。

2. 本書の対象読者

本書は WinActor 向けのシナリオを開発するシナリオ開発者を読者として想定したマニュアルです。ノードを利用した初心者向けの学習シナリオの内容を理解することで、シナリオ開発者に必要な知識を身に付けることができます。また、「操作マニュアル」と合わせてお読み頂けると、WinActor に関する理解が一層深まります。

3. 学習シナリオの動作環境

学習シナリオは WinActor シナリオとして実装されています。

学習シナリオの動作環境は WinActor の動作環境に準じます。

4. 商標について

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

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

5. 本書に関する注意

- 本書および提供するソフトウェア類に付された著作権表示「Copyright © 2013-2025 NTT, Inc. & NTT ADVANCED TECHNOLOGY CORPORATION」の変更、削除をすることはできません。
本書の著作権は NTT 株式会社及び NTT アドバンステクノロジー株式会社に帰属します。
- 本書では、Windows の操作方法や機能を理解されていることを前提として説明しています。本書に記載されていないことについては、Microsoft が提供しているドキュメントなどをご覧ください。

6. 学習シナリオの一覧

学習シナリオの一覧を以下の表に示します。

表 6-1 学習シナリオ一覧

No.	シナリオ名	学習するノード
1	ノード、ライブラリのグループ化	グループ
2	シナリオで付箋追加	-
3	ブザー音発生	音
4	WAVE ファイル音発生	
5	簡単なメッセージ表示	待機ボックス
6	現在の日時取得	日時取得
7	ログイン中のユーザ名取得	ユーザ名取得
8	変数の設定と表示	変数値設定
9	データの入力と表示	インプットボックス
10	データの選択と表示	選択ボックス
11	数値の加算	カウントアップ
12	2つの数字の四則演算	四則演算
13	変数のコピーと表示	変数値コピー
14	半角から全角に変換	全角化・半角化
15	指定時間待機後に続行	指定時間待機
16	クリップボードで値設定・取得	クリップボード
17	2つの選択肢から選択した結果表示	分岐
18	3つ以上の選択肢から選択した結果表示	多分岐
19	操作前判定して繰り返す(条件式)	繰り返し
20	操作前判定して繰り返す(回数)	
21	操作前判定して繰り返す(範囲)	
22	操作前判定して繰り返す(データ数)	
23	操作後判定して繰り返す(条件式)	後判定繰返
24	操作後判定して繰り返す(回数)	
25	操作後判定して繰り返す(範囲)	
26	操作後判定して繰り返す(データ数)	
27	繰り返しを途中で終了	繰り返し終了
28	繰り返しの一部をスキップして次の条件判定に進む	次の条件判定
29	エラーが発生して例外処理の異常系統行	例外処理
30	別のシナリオを呼び出す	シナリオファイル呼び出し

No.	シナリオ名	学習するノード
31	呼び出し先シナリオ	シナリオ終了
32	シナリオからサブルーチンを呼び出す	サブルーチングループ サブルーチン呼び出し サブルーチン終了

7. 各学習シナリオの詳細

各学習シナリオの内容、および使い方を説明します。

7.1. ノード、ライブラリのグループ化

7.1.1. シナリオの機能

2つの数値を足し算し、足し算の結果を表示します。複数のノードをグループ化します。

7.1.2. 事前準備

必要な事前準備はありません。

7.1.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② ユーザに1番目の数値を入力してもらいます。
- ③ ユーザに2番目の数値を入力してもらいます。
- ④ 入力してもらった2つの数値を足し算します。
- ⑤ 足し算の結果を表示します。

7.1.4. 補足情報

- 「グループノード」のユースケース
シナリオの一部の操作、機能、結果を出すために一緒に使ったノードやライブラリをグループとしてまとめたい場合に利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「インプットボックスノード」を利用して足し算するため数値を入力してもらいます。
「四則演算ノード」を利用して入力してもらった2つの数値を足し算します。
「待機ボックスノード」を利用して足し算結果の確認メッセージを表示します。

7.2. シナリオで付箋追加

7.2.1. シナリオの機能

1つの付箋を付けて内容を書きます。

7.2.2. 事前準備

必要な事前準備はありません。

7.2.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② ユーザに表示するデータを入力してもらいます。
- ③ 入力してもらったデータを表示します。

7.2.4. 補足情報

- 「付箋」のユースケース
シナリオに説明、ノート、メモなど書きたい場合は利用できます。
- 「データの入力」付箋の処理内容
「インプットボックスノード」をグループ化して、付箋を追加します。
付箋のタイトル: データの入力
付箋の内容: 表示するデータを入力するグループです。
- シナリオで利用しているそれぞれのノードの処理内容
「インプットボックスノード」を利用して表示するデータを入力してもらいます。
「待機ボックスノード」を利用して入力したデータを表示します。

7.3. ブザー音発生

7.3.1. シナリオの機能

シナリオの実行中にブザー音を発生します。

7.3.2. 事前準備

必要な事前準備はありません。

7.3.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② シナリオを実行するとブザー音を発生します。

7.3.4. 補足情報

- 「音ノード」のユースケース
シナリオ実行中にブザーと自分の WAVE ファイル音を発生したい時やシナリオ実行終了する時にも利用できます。
- 「音ノード」のバリエーション
WAVE ファイル音を発生したい場合は「WAVE ファイル」を選択し、自分の WAVE ファイルを設定して利用できます。

7.4. WAVE ファイル音発生

7.4.1. シナリオの機能

シナリオ実行中に【C:¥Windows¥Media¥tada.wav】の音を発生します。

7.4.2. 事前準備

必要な事前準備はありません。

7.4.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② シナリオを実行すると【C:¥Windows¥Media¥tada.wav】を発生します。

7.4.4. 補足情報

- 「音ノード」のユースケース
シナリオ実行中にブザーと自分の WAVE ファイル音を発生したい時やシナリオ実行終了する時にも利用できます。
- 「音ノード」のバリエーション
ブザー音を発生したい場合は「ブザー」を選択して利用できます。

7.5. 簡単なメッセージ表示

7.5.1. シナリオの機能

「Hello World!」を表示します。

7.5.2. 事前準備

必要な事前準備はありません。

7.5.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② シナリオを実行すると「Hello World!」メッセージを表示します。

7.5.4. 補足情報

- 「待機ボックスノード」のユースケース
データ出力、注意点表示、シナリオ実行を一時的に待機させてデバッグしたい場合は利用できます。
- 「待機ボックスノード」のバリエーション
「待機ボックス」ノードの「表示メッセージ」に「変数名」と「メッセージ」のバリエーションがあります。

7.6. 現在の日時取得

7.6.1. シナリオの機能

「yyyy/mm/dd」日付形式、「OS デフォルト」タイムゾーンで現在の日付と時間を変数に格納して表示します。

例：取得した日付と時間：2024/02/22 14:06:47

7.6.2. 事前準備

必要な事前準備はありません。

7.6.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 「yyyy/mm/dd」日付形式、「OS デフォルト」タイムゾーンで現在の日付と時間を変数に格納します。
- ③ 格納した変数値の確認メッセージを表示します。

7.6.4. 補足情報

- 「日時取得ノード」のユースケース
シナリオの実行時間を記録する場合は利用できます。
- 「日時取得ノード」のバリエーション
「フォーマットタイプ」、「日付形式」、「タイムゾーン」は色々選べるバリエーションがあります。例:「フォーマットタイプ」の場合は、
 - ① 日付と時間
 - ② 日付のみ
 - ③ 時間のみ
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して「日時取得ノード」からもらった日時の確認メッセージを表示します。

7.7. ログイン中のユーザ名取得

7.7.1. シナリオの機能

現在の Window OS にログイン中のユーザ名を変数に格納して表示します。

例：取得したユーザ名：sample

7.7.2. 事前準備

必要な事前準備はありません。

7.7.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② Window OS にログイン中のユーザ名を変数に格納します。
- ③ 格納した変数値の確認メッセージを表示します。

7.7.4. 補足情報

- 「ユーザ名取得ノード」のユースケース。

WinActor の実行ログに実行したユーザ名を記録したい場合に利用できます。

「01_WinActor 制御/08_実行ログ/ログメッセージ出力」を利用し、「ログメッセージ出力」のプロパティにて「出力メッセージ」に変数「取得結果」を設定します。

- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して「ユーザ名取得ノード」からもらったユーザ名の確認メッセージを表示します。

7.8. 変数値設定と表示

7.8.1. シナリオの機能

「Hello World!」を設定して表示します。

7.8.2. 事前準備

2番目の「変数値設定ノード」の設定値に「Hello World!」を設定してください。

7.8.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 1番目の「変数値設定ノード」から変数値を1番目の「待機ボックスノード」に表示します。
1番目の「変数値設定ノード」の変数値が空白であるため、以下のように表示します。
変数値の設定前
変数値:
- ③ 2番目の「変数値設定ノード」から変数値を2番目の「待機ボックスノード」に表示します。
2番目の「変数値設定ノード」の変数値は「Hello World!」に設定されているため、以下のように表示します。
変数値の設定後
変数値:Hello World!

7.8.4. 補足情報

- 「変数値設定ノード」のユースケース
シナリオ実行中に変数値を変更したい場合は利用できます。
- 「変数値設定ノード」と「変数値コピーノード」を一緒に利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して各「変数値設定ノード」に設定した変数値を表示します。

7.9. データの入力と表示

7.9.1. シナリオの機能

入力結果「Hello World!」を表示します。

7.9.2. 事前準備

必要な事前準備はありません。

7.9.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② ユーザに表示するデータ「Hello World!」を入力してもらいます。
- ③ 入力してもらった「Hello World!」を表示します。

7.9.4. 補足情報

- 「インプットボックスノード」のユースケース
変数値を自由に設定してシナリオを実行したい場合は利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して「インプットボックスノード」に設定した変数値「Hello World!」を表示します。

7.10. データの選択と表示

7.10.1. シナリオの機能

選択結果を表示します。

7.10.2. 事前準備

必要な事前準備はありません。

7.10.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② ユーザに表示するデータを以下の選択候補から選択してもらいます。
選択候補：「Hello!」、「World!」
- ③ 選択してもらった結果の確認メッセージを表示します。

7.10.4. 補足情報

- 「選択ボックスノード」のユースケース
事前に選択肢を設定して選択結果によってシナリオを実行したい場合は利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して選択した変数値の確認メッセージを表示します。

7.11. 数値の加算

7.11.1. シナリオの機能

初期値「1」をカウントアップ「+1」して結果「2」を表示します。

7.11.2. 事前準備

必要な事前準備はありません。

7.11.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 「カウンタ」変数の初期値「1」をカウントアップ「+1」します。
- ③ カウントアップした結果「カウンタ」変数値「2」を以下のように表示します。
カウントアップの結果: 2

7.11.4. 補足情報

- 「カウントアップノード」のユースケース
繰り返し(ループ)回を計算する場合は利用できます。
- 「カウントアップノード」のバリエーション
「+1」以上加算のバリエーションがあります。
- 「カウントアップノード」と一緒に利用できるノード提案
シナリオの一部を複数回行う必要がある場合に使用する「繰り返しノード」と「後判定繰返ノード」を一緒に利用できます。
- 「四則演算ノード」の足し算を「カウントアップノード」の代わりに利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してカウントアップした変数値の確認メッセージを表示します。

7.12. 2つの数字の四則演算

7.12.1. シナリオの機能

初期値「1」と「2」を足し算して結果「3」を表示します。

7.12.2. 事前準備

必要な事前準備はありません。

7.12.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 「変数1」の初期値「1」と「変数2」の初期値「2」を足し算します。
- ③ 足し算結果「3」を以下のように表示します。

計算した結果:3

7.12.4. 補足情報

- 「四則演算ノード」のユースケース
数値の演算を行う場合や繰り返し(ループ)の回数を計算する場合に利用できます。
- 「四則演算ノード」のバリエーション
「四則演算」ノードに「+」「-」「×」「÷」のバリエーションがあります。
- 「四則演算ノード」と一緒に利用できるノード提案
シナリオの一部を複数回行う必要がある場合に利用する「繰り返しノード」と「後判定繰返ノード」を一緒に利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して計算結果の確認メッセージを表示します。

7.13. 変数のコピーと表示

7.13.1. シナリオの機能

初期値「Hello World!」をコピーして表示します。

7.13.2. 事前準備

必要な事前準備はありません。

7.13.3. シナリオ実行と処理の流れ

① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。

② コピー前の変数値を表示します。

変数値コピー前、

コピー元:Hello World!

コピー先:

③ 「コピー元」変数の初期値「Hello World!」をコピーします。

④ コピー後の変数値を表示します。

変数値コピー後、

コピー元:Hello World!

コピー先:Hello World!

7.13.4. 補足情報

- 「変数値コピーノード」のユースケース

シナリオを実行する時、変数进行操作する前に変数の元値を保存したい場合は利用できます。

- 「変数値コピーノード」と一緒に利用できるノード提案

「変数値設定ノード」のような変数値を利用して操作するノードと一緒に利用できます。

- シナリオで利用しているそれぞれのノードの処理内容

1 番目「待機ボックスノード」を利用してコピー前の変数値の確認メッセージを表示します。

2 番目「待機ボックスノード」を利用してコピー後の変数値の確認メッセージを表示します。

7.14. 半角から全角に変換

7.14.1. シナリオの機能

初期値「Hello World!」半角を全角化して結果「He l l o W o r l d !」を表示する。

7.14.2. 事前準備

必要な事前準備はありません。

7.14.3. シナリオ実行と処理の流れ

① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。

② 全角化前の変数値を表示します。

全角化前、

変数値:Hello World!

③ 半角変数値「Hello World!」を全角化します。

④ 全角化前の変数値を表示します。

全角化後、

変数値:He l l o W o r l d !

7.14.4. 補足情報

- 「全角化／半角化ノード」のユースケース

変数に設定された数値・文字を、全角または半角に変更したい場合は利用できます。

- 「全角化／半角化ノード」と一緒に利用できるノード提案

「変数値設定ノード」のような変数値を利用して操作するノードと一緒に利用できます。

- シナリオで利用しているそれぞれのノードの処理内容

1 番目「待機ボックスノード」を利用して全角化前の変数値の確認メッセージを表示します。

2 番目「待機ボックスノード」を利用して全角化後の変数値の確認メッセージを表示します。

7.15. 指定時間待機後に実行

7.15.1. シナリオの機能

シナリオを実行して、OK を押すと 5 秒後に「Hello World!」が表示します。

7.15.2. 事前準備

必要な事前準備はありません。

7.15.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 「OK を押すと 5 秒後に「Hello World!」が表示します。」というお知らせメッセージが表示します。
- ③ 「OK」を押してください。
- ④ 5 秒後に「Hello World!」メッセージが表示します。

7.15.4. 補足情報

- 「指定時間待機ノード」のユースケース
シナリオ実行中に一時的に停止して待ちたい場合に利用できます。
- 「指定時間待機ノード」のバリエーション
「指定時間待機」ノードで「指定した時間待つ」、「指定した時刻まで待つ」、「指定時間チェック」のバリエーションがあります。
- シナリオで利用しているそれぞれのノードの処理内容
 - 1 番目「待機ボックスノード」を利用してお知らせメッセージを表示します。
 - 2 番目「待機ボックスノード」を利用して「Hello World!」メッセージを表示します。

7.16. クリップボードで値設定・取得

7.16.1. シナリオの機能

初期値「Hello World!」をクリップボードへ設定し、設定した値をクリップボードから取得してメッセージを表示します。

7.16.2. 事前準備

必要な事前準備はありません。

7.16.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② クリップボードへ値「Hello World!」を設定します。
- ③ クリップボードの値「Hello World!」を取得します。
- ④ 取得した変数値「Hello World!」を表示します。
クリップボードから取得した値:Hello World!

7.16.4. 補足情報

- 「クリップボードノード」のユースケース
クリップボードを介して値のやり取りを行いたい場合に使用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してクリップボードから取得した変数値の確認メッセージを表示します。

7.17. 2つの選択枝から選択した結果表示

7.17.1. シナリオの機能

選択枝「True」、「False」の中から選択された結果を「分岐ノード」で判定して結果を表示します。

7.17.2. 事前準備

必要な事前準備はありません。

7.17.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 「分岐ノード」で判定するデータを以下の選択候補から選択してもらいます。
選択候補：「True」、「False」
- ③ 選択してもらった結果を判定します。
- ④ 判定した結果の確認メッセージを表示します。
Trueを選択した場合は、「True を選びました。」を表示します。
False を選択した場合は、「False を選びました。」を表示します。

7.17.4. 補足情報

- 「分岐ノード」のユースケース
2つ変数値の条件判定によって実行する場合は利用できます。
- 「分岐ノード」のバリエーション
等しい、等しくない、=、≠、>、<、≧、≦、が true、が false、等しい(曖昧)、正規表現の詳細を「操作マニュアル」の「比較演算子」表で確認できます。
- 「分岐ノード」と一緒に利用できるノード提案
「選択ボックスノード」のようなデータを選択できるノードと一緒に利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して選択した変数値の確認メッセージを表示します。

7.18. 3 つ以上の選択肢から選択した結果表示

7.18.1. シナリオの機能

選択肢「1」、「2」、「3」、「以外」の中から選択された結果を「多分岐ノード」で判定して結果を表示します。

7.18.2. 事前準備

必要な事前準備はありません。

7.18.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 「多分岐ノード」で判定するデータを以下の選択候補から選択してもらいます。
選択候補：「1」、「2」、「3」、「以外」
- ③ 選択してもらった結果を判定します。
- ④ 判定した結果の確認メッセージを表示します。
1を選択した場合は、「1を選びました。」を表示します。
2を選択した場合は、「2を選びました。」を表示します。
3を選択した場合は、「3を選びました。」を表示します。
「以外」を選択した場合は、「以外を選びました。」を表示します。

7.18.4. 補足情報

- 「多分岐ノード」のユースケース
3つ以上の変数値の条件判定によって実行する場合は利用できます。
- 「多分岐ノード」のバリエーション
等しい、等しくない、＝、≠、>、<、≧、≦、が true、が false、等しい(曖昧)、正規表現の詳細を「操作マニュアル」の「比較演算子」表で確認できます。
- 「多分岐ノード」と一緒に利用できるノード提案
「選択ボックスノード」のようなデータを選択できるノードと一緒に利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用して選択した変数値の確認メッセージを表示します。

7.19. 操作前判定して繰り返す(条件式)

7.19.1. シナリオの機能

カウンタ値を表示してカウントアップする処理を「5 回」繰り返します。

以下のカウンタ値の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」
- 2 回目の繰り返しの場合は、「2 回目のカウンタ値:2」
- 3 回目の繰り返しの場合は、「3 回目のカウンタ値:3」
- 4 回目の繰り返しの場合は、「4 回目のカウンタ値:4」
- 5 回目の繰り返しの場合は、「5 回目のカウンタ値:5」

7.19.2. 事前準備

必要な事前準備はありません。

7.19.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 以下の間繰り返します。
「繰り返しノード」の条件式に「カウンタ値 \leq 5」を設定しております。
- ③ 各繰り返しに、次のように実行します。
 - 1. 「繰り返し」のカウンタ値を表示します。
 - 2. 「繰り返し」のカウンタ値を「+1」します。

7.19.4. 補足情報

- 「繰り返しノード」のユースケース
複数回処理をしたい場合、かつ最初の処理を行う前に繰り返しを判定したい場合に利用できます。
- 「繰り返しノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオでは条件式を利用しております。
- 「繰り返しノード」と一緒に利用できるノード提案
「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。
当シナリオでは「カウントアップ」ノードを利用しております。
- 「後判定繰返ノード」を「繰り返しノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
 - ・「繰り返しノード」は最初の処理を行う前に判定します。
 - ・「後判定繰返ノード」は最後の処理を行った後に判定します。

- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。
「カウントアップノード」を利用してカウンタ値を 1 ずつ加算します。

7.20. 操作前判定して繰り返す(回数)

7.20.1. シナリオの機能

カウンタ値を表示してカウントアップする処理を「5 回」繰り返します。

以下のカウンタ値の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」
- 2 回目の繰り返しの場合は、「2 回目のカウンタ値:2」
- 3 回目の繰り返しの場合は、「3 回目のカウンタ値:3」
- 4 回目の繰り返しの場合は、「4 回目のカウンタ値:4」
- 5 回目の繰り返しの場合は、「5 回目のカウンタ値:5」

7.20.2. 事前準備

必要な事前準備はありません。

7.20.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 以下の間繰り返します。
「繰り返しノード」の回数の値に「5」を設定しております。
- ③ 各繰り返しに、次のように実行します。
 - 1. 「繰り返し」のカウンタ値を表示します。
 - 2. 「繰り返し」のカウンタ値を「+1」します。

7.20.4. 補足情報

- 「繰り返しノード」のユースケース
複数回処理をしたい場合、かつ最初の処理を行う前に繰り返しを判定したい場合に利用できます。
- 「繰り返しノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオでは回数を利用しております。
- 「繰り返しノード」と一緒に利用できるノード提案
「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。
当シナリオでは「カウントアップ」ノードを利用しております。
- 「後判定繰返ノード」を「繰り返しノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
「繰り返しノード」は最初の処理を行う前に判定します。
「後判定繰返ノード」は最後の処理を行った後に判定します。

- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。
「カウントアップノード」を利用してカウンタ値を 1 ずつ加算します。

7.21. 操作前判定して繰り返す(範囲)

7.21.1. シナリオの機能

カウンタ値を表示してカウントアップする処理を「5 回」繰り返します。

以下のカウンタ値の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」
- 2 回目の繰り返しの場合は、「2 回目のカウンタ値:2」
- 3 回目の繰り返しの場合は、「3 回目のカウンタ値:3」
- 4 回目の繰り返しの場合は、「4 回目のカウンタ値:4」
- 5 回目の繰り返しの場合は、「5 回目のカウンタ値:5」

7.21.2. 事前準備

必要な事前準備はありません。

7.21.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 以下の間繰り返します。
「繰り返しノード」の範囲の左値に「1」、右値に「5」を設定しております。
- ③ 各繰り返しに、次のように実行します。
 - 1. 「繰り返し」のカウンタ値を表示します。
 - 2. 「繰り返し」のカウンタ値を「+1」します。

7.21.4. 補足情報

- 「繰り返しノード」のユースケース
複数回処理をしたい場合、かつ最初の処理を行う前に繰り返しを判定したい場合に利用できます。
- 「繰り返しノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオでは範囲を利用しております。
- 「繰り返しノード」と一緒に利用できるノード提案
「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。
当シナリオでは「カウントアップ」ノードを利用しております。
- 「後判定繰返ノード」を「繰り返しノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
「繰り返しノード」は最初の処理を行う前に判定します。
「後判定繰返ノード」は最後の処理を行った後に判定します。

- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。
「カウントアップノード」を利用してカウンタ値を 1 ずつ加算します。

7.2.2. 操作前判定して繰り返す(データ数)

7.2.2.1. シナリオの機能

「22_NumberOfData_Pre-TestLoop.xlsx」の各行の値を表示する処理を「5 回」繰り返します。

以下の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 行目の値:1」
- 2 回目の繰り返しの場合は、「2 行目の値:2」
- 3 回目の繰り返しの場合は、「3 行目の値:3」
- 4 回目の繰り返しの場合は、「4 行目の値:4」
- 5 回目の繰り返しの場合は、「5 行目の値:5」

7.2.2.2. 事前準備

変数一覧の「データファイル」変数の初期値に「22_NumberOfData_Pre-TestLoop.xlsx」の絶対パスを指定してください。

7.2.2.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 繰り返しの回数は「22_NumberOfData_Pre-TestLoop.xlsx」の行数と等しくなり、「5 回」繰り返します。
「繰り返しノード」のデータ数のデータファイル名に変数「データファイル」を設定しております。
- ③ 各繰り返しに、「22_NumberOfData_Pre-TestLoop.xlsx」の各行の値を表示します。

7.2.2.4. 補足情報

- 「繰り返しノード」のユースケース
データファイルに記入されている複数のデータに対して、繰り返し処理をしたい場合、かつ最初の処理を行う前に繰り返しを判定したい場合に利用できます。
- 「繰り返しノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオではデータ数を利用しております。
- データ数のデータファイル名に指定された Excel ファイルか CSV ファイルの列名と同じ変数名の変数が変数一覧に登録されている場合、繰り返しの回数は値が記載されている行数と等しくなり、自動的に各行の値が各変数に読み込まれます。
- 「繰り返しノード」と一緒に利用できるノード提案

「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。

- 「後判定繰返ノード」を「繰り返しノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
 - 「繰り返しノード」は最初の処理を行う前に判定します。
 - 「後判定繰返ノード」は最後の処理を行った後に判定します。
- シナリオで利用しているそれぞれのノードの処理内容
 - 「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。

7.2.3. 操作後判定して繰り返す(条件式)

7.2.3.1. シナリオの機能

カウンタ値を表示してカウントアップする処理を「5 回」繰り返します。

以下のカウンタ値の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」
- 2 回目の繰り返しの場合は、「2 回目のカウンタ値:2」
- 3 回目の繰り返しの場合は、「3 回目のカウンタ値:3」
- 4 回目の繰り返しの場合は、「4 回目のカウンタ値:4」
- 5 回目の繰り返しの場合は、「5 回目のカウンタ値:5」

7.2.3.2. 事前準備

必要な事前準備はありません。

7.2.3.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 以下の間繰り返します。
「後判定繰返ノード」の条件式に「カウンタ値 \leq 5」を設定しております。
- ③ 各繰り返しに、次のように実行します。
 - 1. 「後判定繰返」のカウンタ値を表示します。
 - 2. 「後判定繰返」のカウンタ値を「+1」します。

7.2.3.4. 補足情報

- 「後判定繰返ノード」のユースケース
複数回処理をしたい場合、かつ最後の処理を行った後に繰り返しを判定したい場合に利用できます。
- 「後判定繰返ノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオでは条件式を利用しております。
- 「後判定繰返ノード」と一緒に利用できるノード提案
「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。
当シナリオでは「四則演算」ノードを利用しております。
- 「繰り返しノード」を「後判定繰返ノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
 - ・「繰り返しノード」は最初の処理を行う前に判定します。
 - ・「後判定繰返ノード」は最後の処理を行った後に判定します。

- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。
「四則演算ノード」を利用してカウンタ値を 1 ずつ加算します。

7.24. 操作後判定して繰り返す(回数)

7.24.1. シナリオの機能

カウンタ値を表示してカウントアップする処理を「5 回」繰り返します。

以下のカウンタ値の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」
- 2 回目の繰り返しの場合は、「2 回目のカウンタ値:2」
- 3 回目の繰り返しの場合は、「3 回目のカウンタ値:3」
- 4 回目の繰り返しの場合は、「4 回目のカウンタ値:4」
- 5 回目の繰り返しの場合は、「5 回目のカウンタ値:5」

7.24.2. 事前準備

必要な事前準備はありません。

7.24.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 以下の間繰り返します。
「後判定繰返ノード」の回数の値に「5」を設定しております。
- ③ 各繰り返しに、次のように実行します。
 - 1. 「後判定繰返」のカウンタ値を表示します。
 - 2. 「後判定繰返」のカウンタ値を「+1」します。

7.24.4. 補足情報

- 「後判定繰返ノード」のユースケース
複数回処理をしたい場合、かつ最後の処理を行った後に繰り返しを判定したい場合に利用できます。
- 「後判定繰返ノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオでは回数を利用しております。
- 「後判定繰返ノード」と一緒に利用できるノード提案
「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。
当シナリオでは「四則演算」ノードを利用しております。
- 「繰り返しノード」を「後判定繰返ノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
「繰り返しノード」は最初の処理を行う前に判定します。
「後判定繰返ノード」は最後の処理を行った後に判定します。

- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。
「四則演算ノード」を利用してカウンタ値を 1 ずつ加算します。

7.25. 操作後判定して繰り返す(範囲)

7.25.1. シナリオの機能

カウンタ値を表示してカウントアップする処理を「5 回」繰り返します。

以下のカウンタ値の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」
- 2 回目の繰り返しの場合は、「2 回目のカウンタ値:2」
- 3 回目の繰り返しの場合は、「3 回目のカウンタ値:3」
- 4 回目の繰り返しの場合は、「4 回目のカウンタ値:4」
- 5 回目の繰り返しの場合は、「5 回目のカウンタ値:5」

7.25.2. 事前準備

必要な事前準備はありません。

7.25.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 以下の間繰り返します。
「後判定繰返ノード」の範囲の左値に「1」、右値に「5」を設定しております。
- ③ 各繰り返しに、次のように実行します。
 - 1. 「後判定繰返」のカウンタ値を表示します。
 - 2. 「後判定繰返」のカウンタ値を「+1」します。

7.25.4. 補足情報

- 「後判定繰返ノード」のユースケース
複数回処理をしたい場合、かつ最後の処理を行った後に繰り返しを判定したい場合に利用できます。
- 「後判定繰返ノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオでは範囲を利用しております。
- 「後判定繰返ノード」と一緒に利用できるノード提案
「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。
当シナリオでは「四則演算」ノードを利用しております。
- 「繰り返しノード」を「後判定繰返ノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
「繰り返しノード」は最初の処理を行う前に判定します。
「後判定繰返ノード」は最後の処理を行った後に判定します。

- シナリオで利用しているそれぞれのノードの処理内容
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。
「四則演算ノード」を利用してカウンタ値を 1 ずつ加算します。

7.26. 操作後判定して繰り返す(データ数)

7.26.1. シナリオの機能

「26_NumberOfData_Post-TestLoop.xlsx」の各行の値を表示する処理を「5 回」繰り返します。

以下の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 行目の値:1」
- 2 回目の繰り返しの場合は、「2 行目の値:2」
- 3 回目の繰り返しの場合は、「3 行目の値:3」
- 4 回目の繰り返しの場合は、「4 行目の値:4」
- 5 回目の繰り返しの場合は、「5 行目の値:5」

7.26.2. 事前準備

変数一覧の「データファイル」変数の初期値に「26_NumberOfData_Post-TestLoop.xlsx」の絶対パスを指定してください。

7.26.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 後判定繰返の回数は「26_NumberOfData_Post-TestLoop.xlsx」の行数と等しくなり、「5 回」繰り返します。
「後判定繰返ノード」のデータ数のデータファイル名に変数「データファイル」を設定しております。
- ③ 各繰り返しに、「26_NumberOfData_Post-TestLoop.xlsx」の各行の値を表示します。

7.26.4. 補足情報

- 「後判定繰返ノード」のユースケース
複数回処理をしたい場合、かつ最後の処理を行った後に繰り返しを判定したい場合に利用できます。
- 「後判定繰返ノード」のバリエーション
条件式、回数、範囲、データ数、データ数(DB 連携)
当シナリオではデータ数を利用しております。
- データ数のデータファイル名に指定された Excel ファイルか CSV ファイルの列名と同じ変数名の変数が変数一覧に登録されている場合、繰り返しの回数は値が記載されている行数と等しくなり、自動的に各行の値が各変数に読み込まれます。
- 「後判定繰返ノード」と一緒に利用できるノード提案

「カウントアップ」、「四則演算」を利用して繰り返し(ループ)回を計算できます。

- 「繰り返しノード」を「後判定繰返ノード」の代わりに利用できます。
- 「繰り返し」と「後判定繰返」の違い
 - 「繰り返しノード」は最初の処理を行う前に判定します。
 - 「後判定繰返ノード」は最後の処理を行った後に判定します。
- シナリオで利用しているそれぞれのノードの処理内容
 - 「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。

7.27. 繰り返しを途中で終了

7.27.1. シナリオの機能

繰り返しを5回設定したが1回だけ実行して終了します。

以下のカウンタ値を1回のみ表示します。

1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」

7.27.2. 事前準備

必要な事前準備はありません。

7.27.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 以下の間繰り返すように「繰り返しノード」の条件式に「カウンタ値 \leq 5」を設定しております。
- ③ 1 回目の繰り返しに、次のように実行します。
 1. 「繰り返し」のカウンタ値を表示します。
 2. 「繰り返し」のカウンタ値を「+1」します。
 3. 「繰り返し」を1回だけ実行して「繰り返し」を終了します。

7.27.4. 補足情報

- 「繰り返し終了ノード」のユースケース
「分岐」ノードなどを利用して繰り返しを途中で停止したい場合に利用できます。
- 「繰り返し終了ノード」と一緒に利用できるノード提案
「繰り返し」、「後判定繰返」を利用して繰り返すことができます。
「分岐」、「多分岐」を利用して選択肢によって判定できます。
当シナリオでは「繰り返し」ノードを利用しております。
- シナリオで利用しているそれぞれのノードの処理内容
「繰り返しノード」を利用してカウンタ値の通り繰り返します。
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。
「カウントアップノード」を利用してカウンタ値を 1 ずつ加算します。

7.28. 繰り返しの一部をスキップして次の条件判定に進む

7.28.1. シナリオの機能

繰り返しを5回設定し、カウンタ値を表示するが3回目はスキップします。

以下の確認メッセージを表示します。

- 1 回目の繰り返しの場合は、「1 回目のカウンタ値:1」
- 2 回目の繰り返しの場合は、「2 回目のカウンタ値:2」
- 4 回目の繰り返しの場合は、「4 回目のカウンタ値:4」
- 5 回目の繰り返しの場合は、「5 回目のカウンタ値:5」

7.28.2. 事前準備

必要な事前準備はありません。

7.28.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② カウンタ値が 5 未満の間繰り返すように「繰り返しノード」の条件式に「カウンタ値<5」を設定しております。当シナリオではカウンタ値を0とします。
- ③ 各繰り返しに、次のように実行します。
 1. 「繰り返し」のカウンタ値を「+1」します。
 2. カウンタ値が 3 に等しいかどうかをチェックします。
 - カウンタ値が 3 に等しい場合は、カウンタ値を表示しないで次の条件にスキップします。
 - カウンタ値が 3 以外の場合は、カウンタ値を表示します。

7.28.4. 補足情報

- 「次の条件判定ノード」のユースケース
条件によって繰り返し回をスキップして次回を実行したい場合は利用できます。
- 「次の条件判定ノード」と一緒に利用できるノード提案
「繰り返し」、「後判定繰返」を利用して繰り返すことができます。
「分岐」、「多分岐」を利用して選択肢によって判定できます。
当シナリオでは「繰り返し」と「分岐」を利用しております。
- シナリオで利用しているそれぞれのノードの処理内容
「繰り返しノード」を利用してカウンタ値の通り繰り返します。
「カウントアップノード」を利用してカウンタ値を 1 ずつ加算します。
「分岐ノード」を利用してスキップする条件を判定します。
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します。

7.29. エラーが発生して例外処理の異常系統行

7.29.1. シナリオの機能

エラー発生時の例外処理を実行します。

今回は 10 を 0 で割ることでエラーを発生させて、以下のエラー確認メッセージを表示します。

「アクション例外処理です。数値を 0 で割ることはできません。」

7.29.2. 事前準備

必要な事前準備はありません。

7.29.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 10 を 0 で割ります。
- ③ エラーを発生させてエラー確認メッセージを表示します。

7.29.4. 補足情報

- 「例外処理ノード」のユースケース
シナリオを実行中にエラーになる場合は、シナリオを停止しないで異常系の例外処理でシナリオを続けるため利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
「四則演算ノード」を利用して 10 を 0 で割ります。
「待機ボックスノード」を利用してエラー確認メッセージを表示します。

7.30. 別のシナリオを呼び出す

7.30.1. シナリオの機能

全角「He l l o W o r l d !」を設定して「31_呼び出し先シナリオ」シナリオを呼び出します。

「31_呼び出し先シナリオ」シナリオから戻した半角化結果「Hello World!」を表示します。

【30_別のシナリオを呼び出す】と【31_呼び出し先シナリオ】はセットで実行する必要があります。

7.30.2. 事前準備

必要な事前準備はありません。

7.30.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 【31_呼び出し先シナリオ】を呼び出し前の確認メッセージを表示します。
全角変数値:He l l o W o r l d !
- ③ 【31_呼び出し先シナリオ】を呼び出します。
- ④ 【31_呼び出し先シナリオ】を呼び出し後の確認メッセージを表示します。
半角化された変数値:Hello World!

7.30.4. 補足情報

- 「シナリオファイル呼び出しノード」のユースケース
色々なシナリオから呼び出せる共通シナリオを呼び出したい場合は利用できます。
- シナリオで利用しているそれぞれのノードの処理内容
 - 1 番目「待機ボックスノード」を利用して【31_呼び出し先シナリオ】を呼び出し前の全角変数値の確認メッセージを表示します。
 - 2 番目「待機ボックスノード」を利用して【31_呼び出し先シナリオ】を呼び出し後の半角化された変数値の確認メッセージを表示します。

7.31. 呼び出し先シナリオ

7.31.1. シナリオの機能

【30_別のシナリオを呼び出す】シナリオからもらった全角変数値「H e l l o W o r l d !」を

半角化させた結果「Hello World!」を戻します。

共通シナリオで操作された変数値を元のシナリオに戻すために使用できます。

7.31.2. 事前準備

必要な事前準備はありません。

7.31.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② 【30_別のシナリオを呼び出す】シナリオからもらった全角変数値「H e l l o W o r l d !」を半角化します。
- ③ シナリオを終了して半角化された変数値「Hello World!」を戻します。

7.31.4. 補足情報

- シナリオで利用しているそれぞれのノードの処理内容
「全角化/半角化ノード」を利用して【30_別のシナリオを呼び出す】シナリオからもらった全角変数値「H e l l o W o r l d !」を半角化します。

7.32. シナリオからサブルーチンを呼び出す

7.32.1. シナリオの機能

シナリオから「サブルーチングループ」を「5 回」呼び出して以下のカウンタ値の確認メッセージを表示します。

- 1 回目のサブルーチングループを呼び出しの場合は、「1 回目のカウンタ値:1」
- 2 回目のサブルーチングループを呼び出しの場合は、「2 回目のカウンタ値:2」
- 3 回目のサブルーチングループを呼び出しの場合は、「3 回目のカウンタ値:3」
- 4 回目のサブルーチングループを呼び出しの場合は、「4 回目のカウンタ値:4」
- 5 回目のサブルーチングループを呼び出しの場合は、「5 回目のカウンタ値:5」

7.32.2. 事前準備

必要な事前準備はありません。

7.32.3. シナリオ実行と処理の流れ

- ① シナリオ実行ボタンをクリック、あるいはキーボードで「F5」を入力します。
- ② シナリオで「繰り返し」から「サブルーチングループ」を「5 回」呼び出します。
「サブルーチングループ」が呼び出されるたびに、次のように実行します。
 - 1. シナリオで「繰り返し」のカウンタ値を表示します。
 - 2. 「サブルーチングループ」で繰り返しのカウンタ値を「+1」します。
 - 3. 「サブルーチングループ」からシナリオに戻ります。

7.32.4. 補足情報

- 「サブルーチン」のユースケース
同じ処理を何度も行いたい場合、その処理をサブルーチン化することで処理を使いまわすことができ、かつシナリオも見やすくなります。
- 当シナリオでは「サブルーチン呼び出し」、「サブルーチングループ」、「サブルーチン終了」を一緒に利用しておりますが、「サブルーチン終了」を利用しないでも「サブルーチン」の操作が終わったらメインシナリオに戻れます。
- シナリオで利用しているそれぞれのノードの処理内容
「繰り返しノード」を利用してカウンタ値が 5 以下の間、繰り返します。
「カウントアップノード」を利用してカウンタ値を 1 ずつ加算します。
「待機ボックスノード」を利用してカウンタ値の確認メッセージを表示します



NTTアドバンステクノロジー株式会社

Copyright © 2013-2025 NTT, Inc. & NTT ADVANCED TECHNOLOGY CORPORATION

本書は著作権法上の保護を受けています。本書の一部あるいは全部を無断で複写、複製することは禁じられています。

本マニュアルの内容は予告なく変更される場合があります。

WA7-AG-20250603
