

RPA Community

XPath入門講座 初歩から関数まで

～変化に強いシナリオ作りのコツ～

2023年 2月 28日

NTTアドバンステクノロジー株式会社

1章: XPathとは？

2章: WinActorでどう使うの？

3章: WinActorでXPathの値が取れないのはどんな時？どうすれば？

4章: XPathで値を取っておけば安心？（より変化に強いXPath活用）

■1章: XPathとは?

■XPath(XML Path Language)とは

XML文書の中の特定の要素や属性を指定して、情報を取得するための言語
同じタグを使用したツリー構造のマークアップ言語であるHTMLにも適用が可能

■HTML(Hyper Text Markup Language)との違い

HTMLは主にウェブサイトのコンテンツの構造を作るために使用される言語

XPathはHTMLに記述されている特定の要素を取得するための言語といえる

```
<html>
<body>
  <h1>演習問題</h1>
  <div class="main">
    <ul>
      <li class="title">XPath</li>
      <li class="title">HTML</li>
      <li class="title">WinActor</li>
    </ul>
  </div>
</body>
</html>
```

左例から”演習問題”の文字を取得するXPath

/html/body/h1

タグ名	概要
html	HTML文書であるという指定
body	文書の本体部分を指定
h1~	見出しを指定
div	ブロック要素を指定
ul	順序のない箇条書きのリストを指定
li	リストの項目を指定

1章：XPathとは？ 【広報的な解説】

XPathとはウェブサイトに散らばるコンテンツ(要素)の位置を知る**住所**です

①画像マッチングが初心者を使いやすい理由は2つ

- ・指差しで指定できる
- ・HTMLの知識が無くても使える
- 地図で例えると
- ・「〇〇ビルの横」など
- ・HTMLは住所録

②画像が(再開発で地図が)変わるとマッチング率低下。画像(地図)を更新する必要がある

③目印のビルが無くなっても目的地の住所は変わらない。それがXPath！ (そういう事にしておいてください。。)

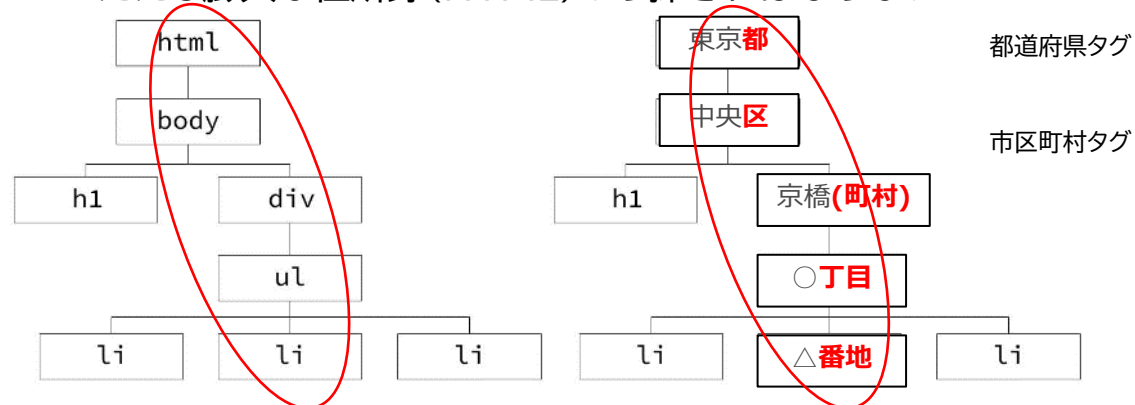
①目印を元に目的地へ！
「青いビルの横で待合せ」



②目印が無くなった！迷子。。
地図を更新しなきゃ。。



③住所なら変わらない
ただし膨大な住所録(HTML)から探さねばならない




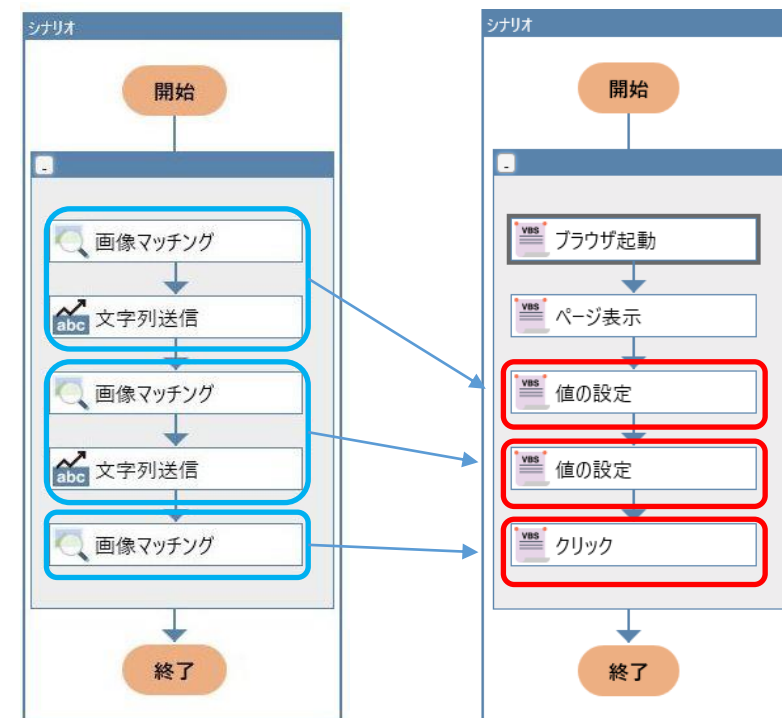
1章: XPathとは?

■ XPathが使えると何が嬉しいの?

・簡単にブラウザ操作ができる ・メンテナンスの手間を少なくできる ・動作が早い

- ・「どこ」の指定が簡単。「どこを」「どこへ」「どこから」で動作指示できるライブラリが充実
- ・同じ操作でも画像マッチングに比べ必要なライブラリ数が減る

目的	画面	No.	ターゲット	画像マッチング	23_ブラウザ関連
ログイン		1.1	[ユーザーID]	TextBoxへカーソル移動	値の設定
		1.2		TextBoxへ指定文字列入力	
		1.3	[パスワード]	TextBoxへカーソル移動	値の設定
		1.4		TextBoxへ指定文字列入力	
		1.5	[ログイン]	ボタンをクリック	クリック



■ XPathが注目されるようになった背景

IEサポート終了によるブラウザ移行、DX促進によるUI変更により
マッチング画像や自動記録アクションの取り直し=メンテナンスが発生

画像に左右されない**変化に強いブラウザ操作方法**として、XPathが使われてます!

■ 2章: WinActorでどう使うの？

■ 「誰でも簡単に使えるWinActor」でHTMLの勉強が必要ななの？

- > 画像マッチングが初心者を使いやすい理由は
- > ・HTMLの知識が無くても使える ・地図感覚で「〇〇ビルの横」など指差しで使える

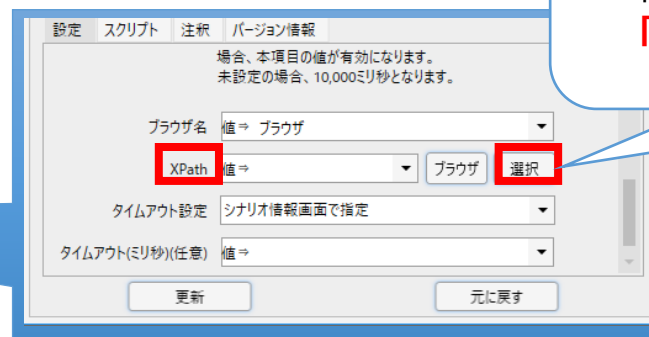
HTMLの知識が無くても使えます！ (この入門編の範囲なら)

XPath取得方法は主に2つ

- ・WinActorの機能で簡単に取得/動作指示できる「23_ブラウザ関連」ライブラリ
- ・ブラウザ の機能で簡単に取得 できる「HTMLを調べるためのツール」



「ボタン」にマウスを重ねると色が変わる => 操作可能！



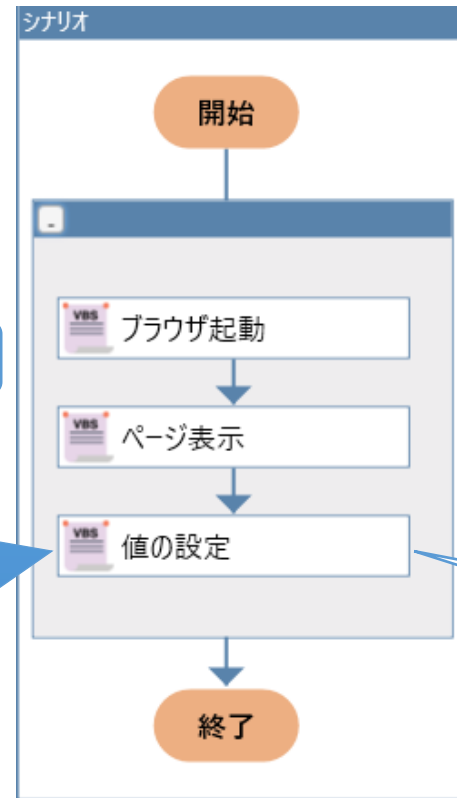
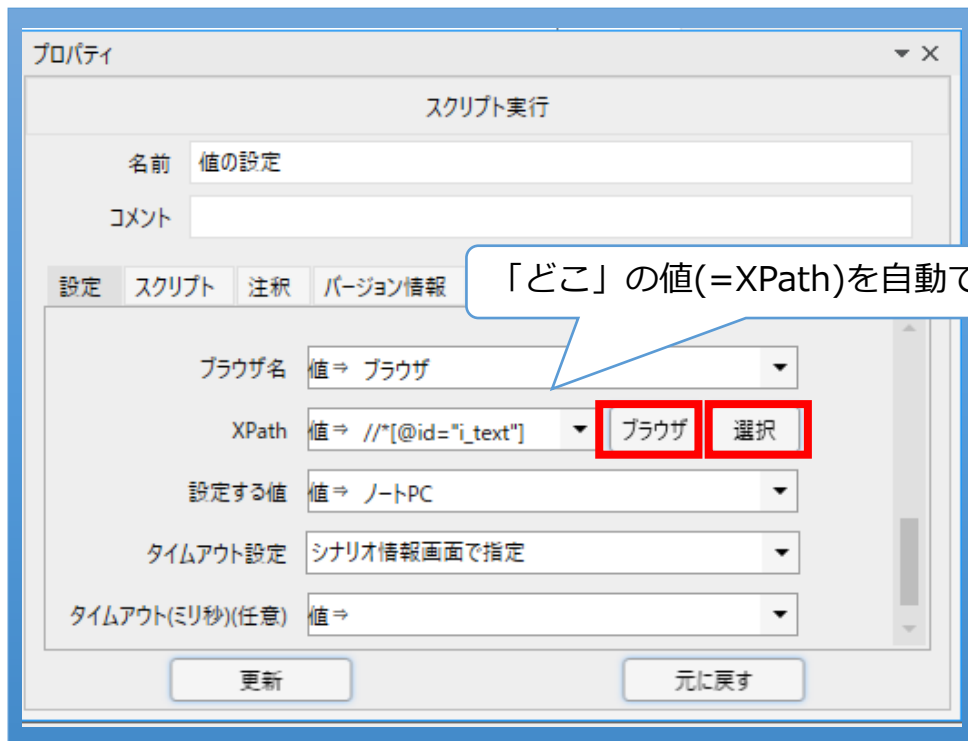
- ・どこをクリックするのか
 - ・どこから文字を取得するのか
 - ・どこに文字を入力するのか
- ⇒ 「どこ」を指定するのがXPath
「どこに〇〇する」を指示するのがライブラリ
難しい「どこ」を調べるのがブラウザツール

2章：「23_ブラウザ関連」ライブラリってどう使うの？ NTTAT

XPathの値の取り方：①プロパティ「XPath：ブラウザ」をクリック
②プロパティ「XPath：選択」をクリック
③ブラウザ上の操作対象をクリック

※ ただし「どのブラウザ」が事前に指定する必要があるので、「ブラウザ起動」「ページ表示」が必要

住所でいえば「日本」



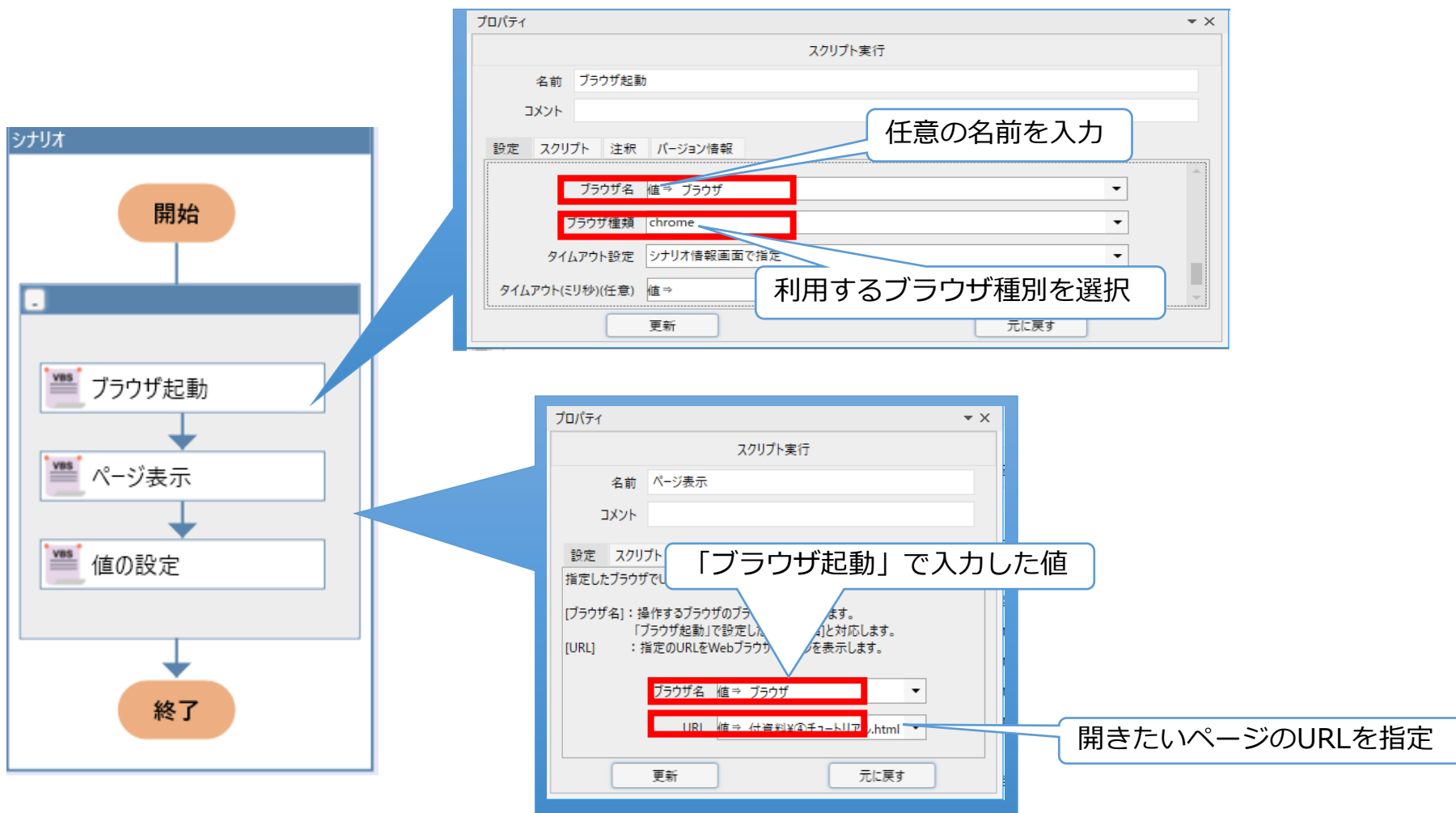
「23_ブラウザ関連」
> 「値の設定」ライブラリ

23_ブラウザ関連 / 01_起動&クローズ / ・ブラウザ起動
23_ブラウザ関連 / 00 / ・ページ表示
23_ブラウザ関連 / 03_クリック / ・クリック

例:「ブラウザの任意の場所に値を入力」してみましょう 1/2 NTTAT

まずは「ブラウザを起動し指定されたURLを開く」 このあたりはいつもどおり

※「ブラウザ名」は後の処理でも指名する必要があります。Ver7.3からは自動的に引き継いでくれます。



シナリオ

開始

ブラウザ起動

ページ表示

値の設定

終了

プロパティ

スクリプト実行

名前 ブラウザ起動

コメント

設定 スクリプト 注釈 バージョン情報

ブラウザ名 値⇒ ブラウザ

ブラウザ種類 chrome

タイムアウト設定 シナリオ情報画面で指定

タイムアウト(ミリ秒)(任意) 値⇒

更新 元に戻す

任意の名前を入力

利用するブラウザ種別を選択

プロパティ

スクリプト実行

名前 ページ表示

コメント

設定 スクリプト

指定したブラウザで

[ブラウザ名]: 操作するブラウザのブラウザ名を指定します。
「ブラウザ起動」で設定したブラウザ名と対応します。

[URL]: 指定のURLをWebブラウザに表示します。

ブラウザ名 値⇒ ブラウザ

URL 値⇒ http://www.ntt.com/html

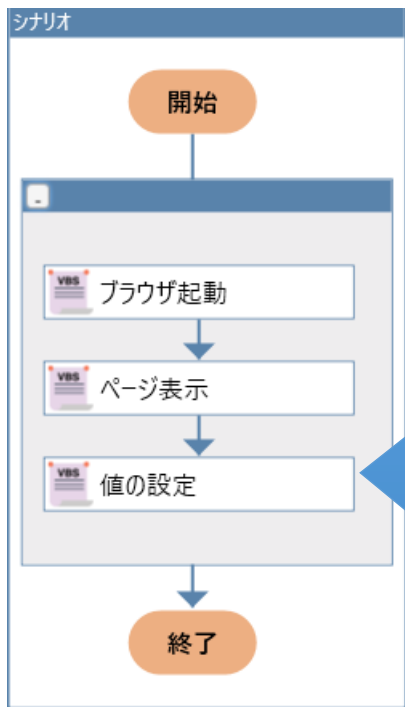
更新 元に戻す

「ブラウザ起動」で入力した値

開きたいページのURLを指定

例:「ブラウザの任意の場所に値を入力」してみましょう 2/2 NTTAT

「ブラウザ」ボタンをクリックすると事前に指定したブラウザ種別で指定したURLが開きます。
「選択」ボタンをクリックしてブラウザ上の操作対象をクリックすると
自動でXPathの値が取得できます。(* Ver7.3以降の機能)



プロパティ

スクリプト実行

名前 値の設定

コメント

設定 スクリプト

④ 自動で取得

ブラウザ名 値 ⇒ ブラウザ ①

XPath 値 ⇒ //*[@id="*_text"] ②

設定する値 値 ⇒ 名古屋

タイムアウト設定 ショート情報画面で指定

タイムアウト(ミリ秒) ⑤ 任意で入力

更新 元に戻す

演習問題

Test

- Xpath
- HTML
- WinActor
- Windows

出発 到着

ボタン

```
<html>
<body>
  <h1>演習問題</h1>
  <div class="main">
    <ul>
      <li>Test</li>
      <li class="title" id="XPath">XPath</li>
      <li class="title" id="HTML">HTML</li>
      <li class="title" id="WinActor">WinActor</li>
      <li class="title" id="Windows">Windows</li>
    </ul>
  </div>
  出発<input type="text" id="query_input" name="from" value="大阪">
  到着<input type="text" id="query_input" name="to" value="東京">
  <BR><BR>
  <input type="button" id="button1" onclick="alert('Test');" value="ボタン">
</body>
</html>
```

■ 3章: WinActorでXPathの値が取れないのはどんな時? どうすれば?

■3章:XPathの値が取れないのはどんな時? どうすれば? NTT AT

①HTMLの作り(構造?)がおかしい場合

HTMLの構成にクセがある場合(誤りとは限らない)、WinActorで値を自動取得が出来ないことがあります。その場合、各種ブラウザに用意された「HTMLを調べるためのツール」でXPathを取得します。

例え 語 :東京都中央区中央区京橋〇丁目 (中央区中央区?)
誤 :東京都中央<京橋〇丁目 (中央<京橋区? 中央<京橋市?)
論外 :とうきょうとちゅうおうくきょうばしまるちょうめ (そもそも住所か?!)

②サイトがフレーム構造の場合

フレーム構造に対してWinActorは自動で(23_ブラウザ関連ライブラリで)値取得が出来ません
開発者ツールで値を取ってもフレームを指定しないと使えません(後述)

例え あなたは「日本」にいます。東京でも大阪でもありません。
日本で取った「北区」の値は「東京」か「大阪」に移動しないと通じません。
東京から大阪に移動する際は日本を経由してください

日本	
東京 ●板橋区 ● 北区 ●豊島区 (東京都北区)	大阪 ●梅田 ● 北区 ●なんば (大阪府大阪市北区)

各種ブラウザに用意された「HTMLを調べるためのツール」でXPathを取得します。

*本稿ではEdgeの「開発者ツール」を用いますがChromeは「検証」、Firefox は「調査」を用います。

開発者ツール(edge)

初回は「ようこそ」とか出ますが「要素」タグを選択

SAMPLE Document

Test

- Xpath
- HTML
- WinActor
- Windows

出発 到着

ボタン

```
<html>
<head></head>
<body == $0
<h1>SAMPLE Document</h1>
<div class="main">...</div>
" 出発"
<input type="text" id="query_input" name="from" value="大阪">
" 到着"
<input type="text" id="query_input" name="to" value="東京">
<br>
<br>
<br>
<input type="button" id="button1" onclick="alert('Test');" value="ボタン">
</body>
</html>
```

html body

スタイル 計算済み レイアウト イベントリスナー DOM ブレークポイント >>

フィルター :hov .cls + [[]

element.style { }

body { ユーザー エージェントのスタイル シート

「要素選択」画面

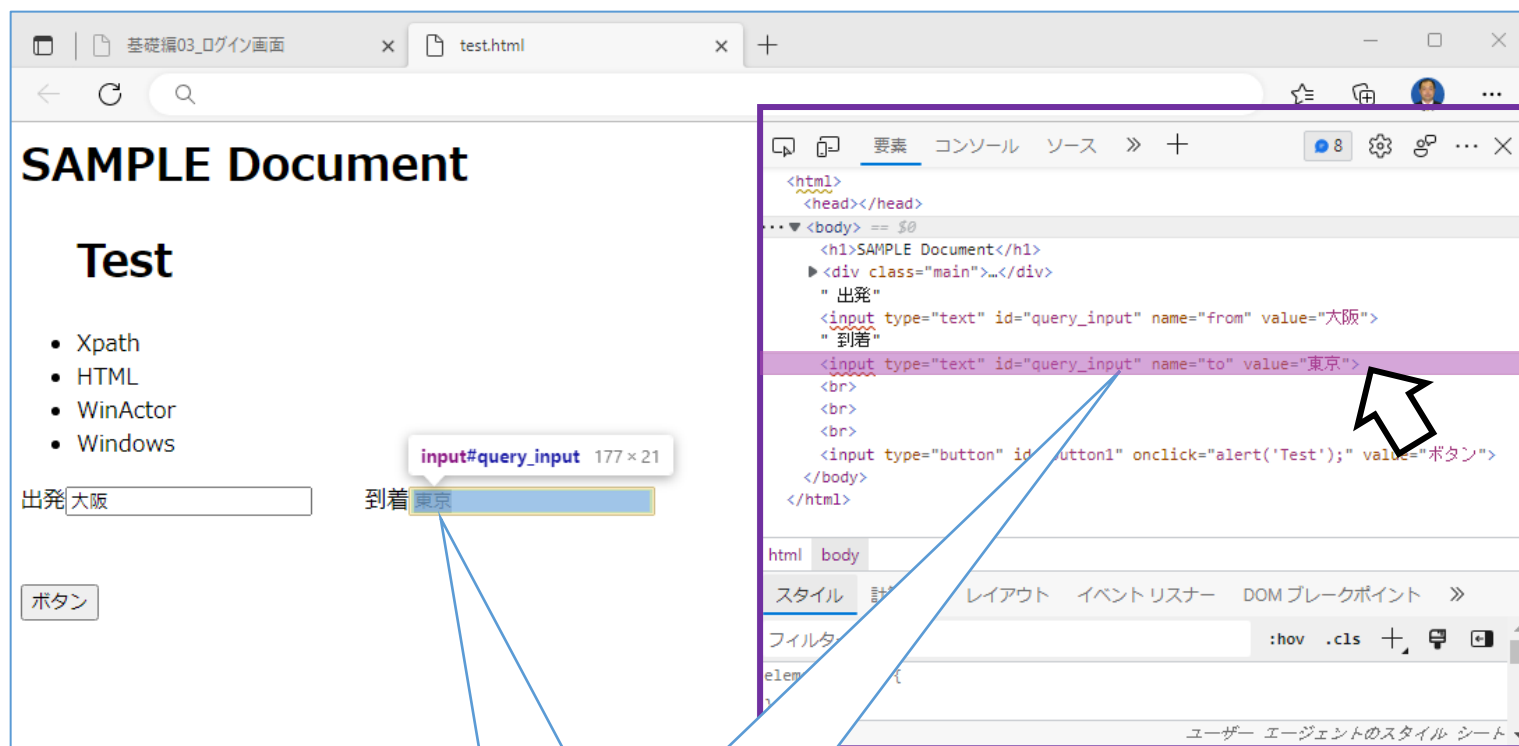
「HTML選択」画面

開発者ツールを用いたXPath取得

使い方 「左で選んで、右で値取得」

STEP 1、ファンクションキー「F12」を押して開発者ツールを起動。

最初は「HTML選択モード」で立ち上がりマウスでHTMLを指すと、対象の要素と共に色が反転します



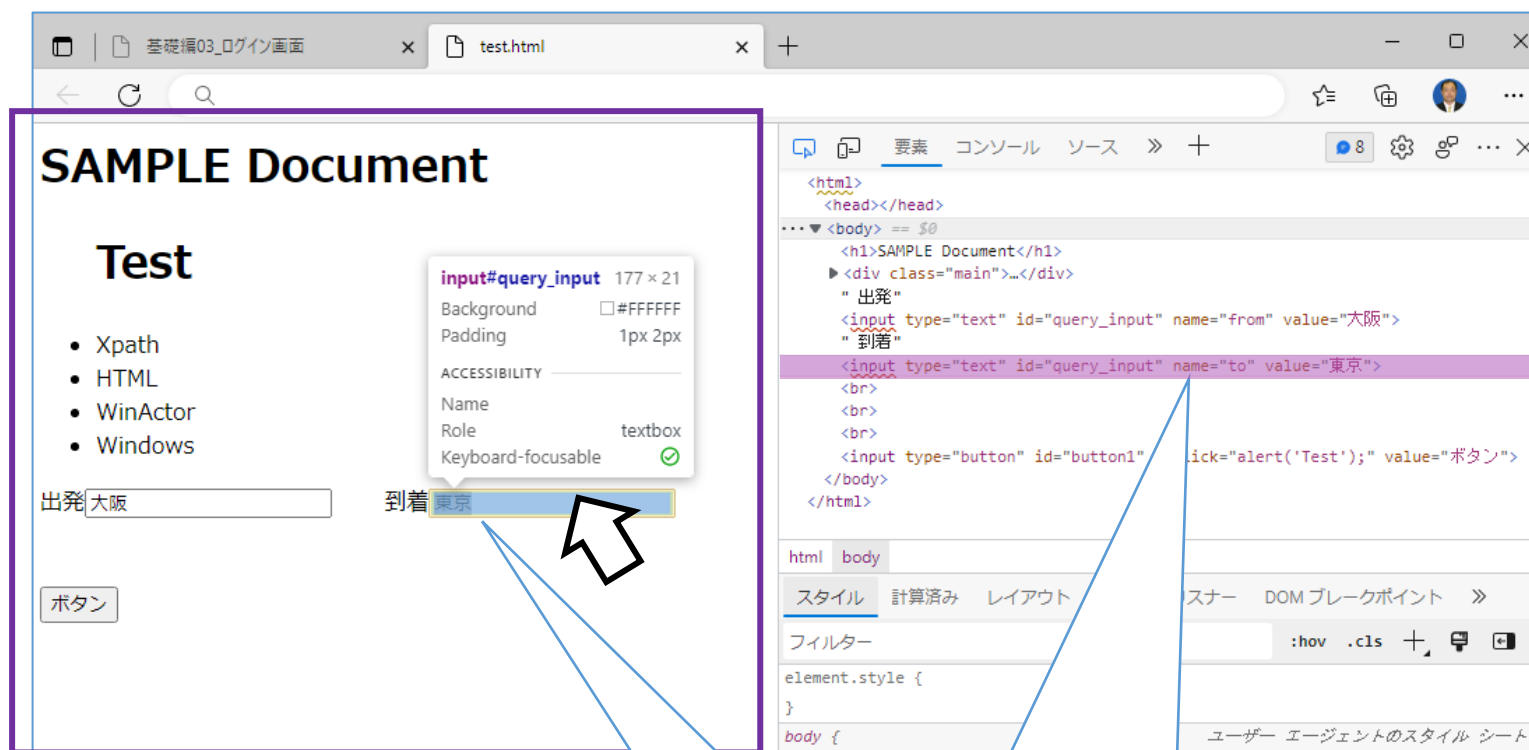
右のHTMLが表示してるのは
左のテキスト入力ボックス

「HTML選択」画面

STEP 2、「Ctrl+Shift+C」同時押しで「要素選択モード」に切り替え。

ブラウザ上で位置を知りたい要素にマウスを合わせると、右画面のHTMLと共に色が反転します
要素上で左クリックすることで、選択を固定しHTML選択モードに切り替わります。

*別の要素を選び直すときは、もう一度「Ctrl+Shift+C」同時押し



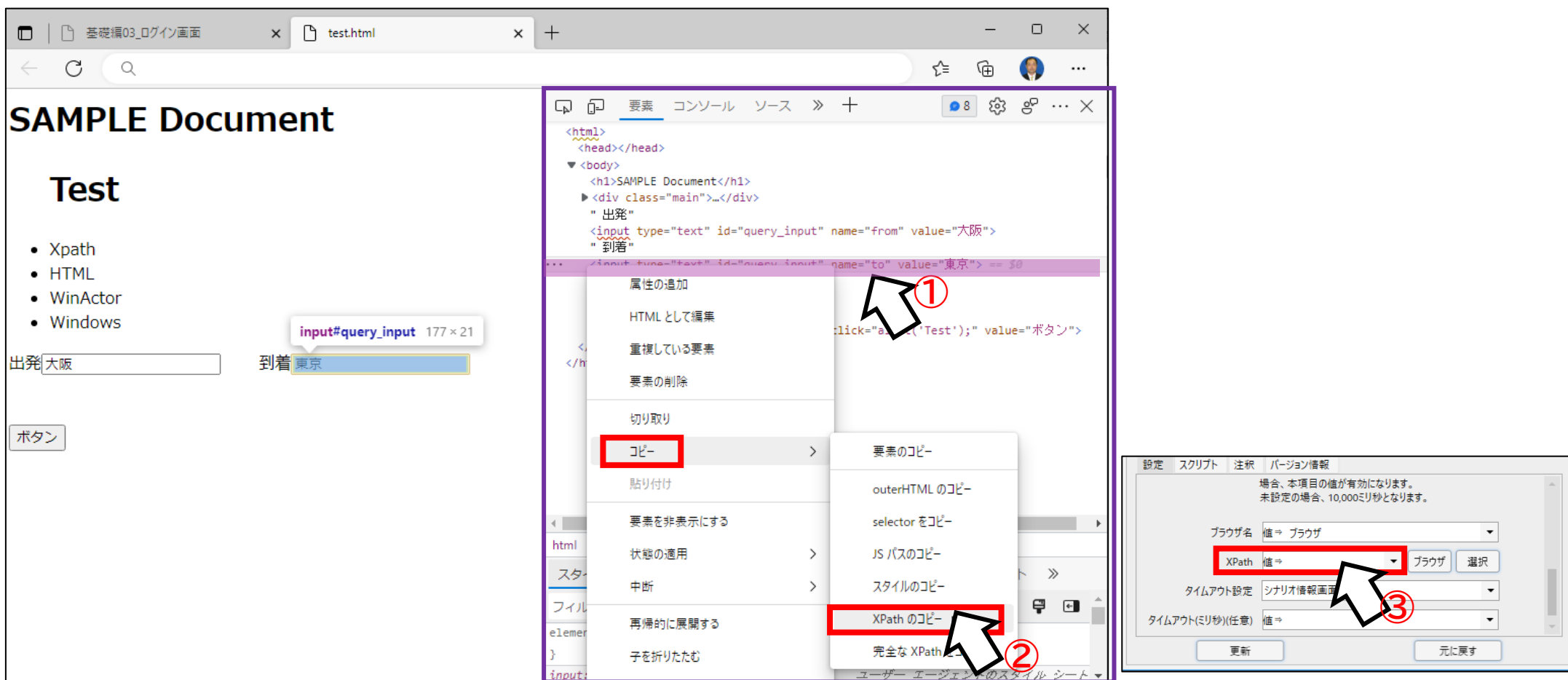
「要素選択」画面

左のマウスが指すテキスト入力ボックスは
右のHTMLが表示している

開発者ツールを用いたXPath取得

STEP 3, HTML選択画面で取得対象を右クリック > コピー > 「XPathのコピー」を左クリック

※この時、下にある「完全なXPathのコピー」を選ばないように注意
得られた値をライブラリのプロパティ「XPath: 値⇒」に貼り付ければ完了です。



XPathの値が取れない例: フレーム構造のとき

フレーム構造に対してWinActorは自動で(23_ブラウザ関連ライブラリで)値取得が出来ません
開発者ツールで値を取っても**フレームを指定しないと使えません**

それはそれとして、

ふつーーーのホームページでWinActorのXPath値取得が上手く行かないことはありませんか？

※プロパティ「XPath>選択」を押してから対象にマウスを重ねても
対象の色が反転せず値が取得できない とか。

問題です

どっちがフレーム構造のホームページでしょうか？

メニュー	興松のホームページ
● トップ	
● 映画出演	
●□□□	
●□□□	
● 出版本	
●□□□	



WinActorシナリオ デモシステム

ユーザーID

パスワード

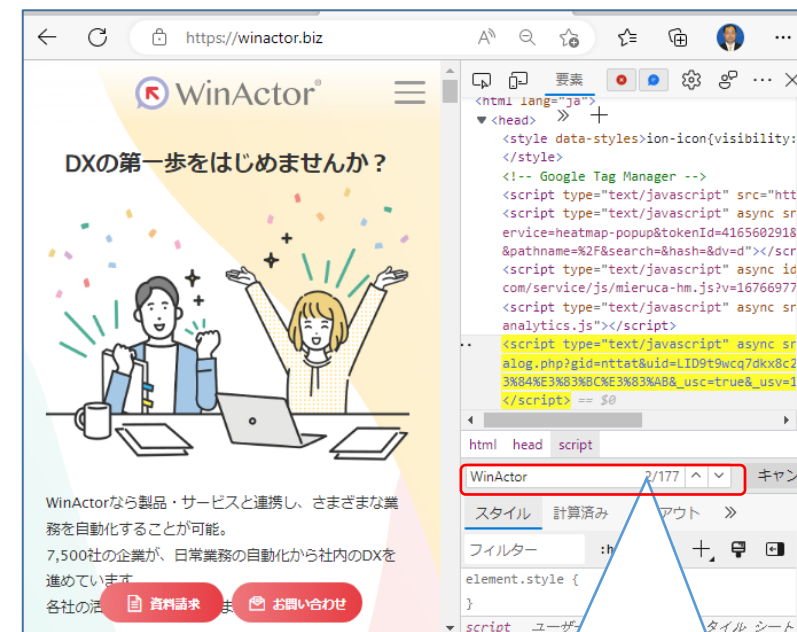
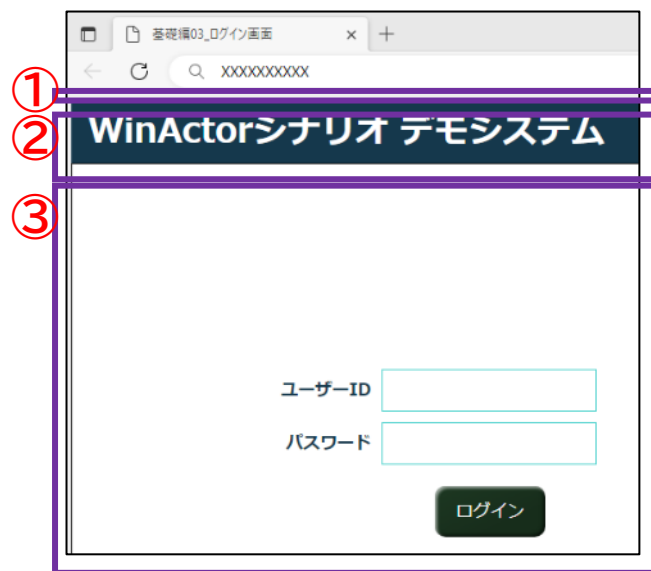
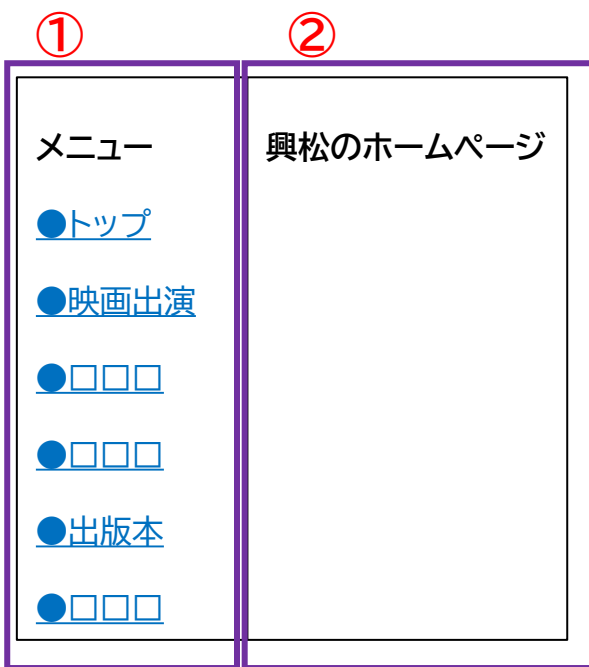
ログイン

XPathの値が取れない例: フレーム構造のとき

答: どちらもフレーム構造でした

おかしいと思ったら開発者ツールで「frame」を検索してみてください。(frame, iframeがあればフレーム構造)

検索窓の右にはヒット数が表示され、上下ボタンで該当箇所(黄色)に移動します。
後述する関数「何番目の要素を指定」でも活用できます。



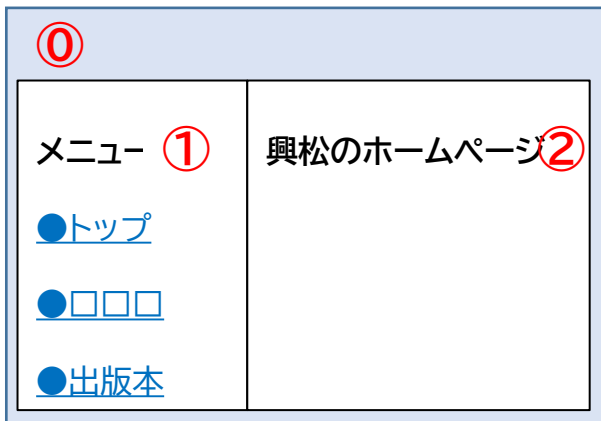
「Ctrl+F」キーで検索窓表示。
「WinActor」が177個ありました

「23_ブラウザ関連/フレーム選択」ライブラリとは

WinActorのブラウザ操作では同時に一つのフレームに対してのみ操作を行えます。
操作対象のフレームを選択するのが「23_ブラウザ関連/フレーム選択」ライブラリです。

以下の注意点にお気を付けください。

- ・フレームが無い一般的なサイトは「0番目のフレーム(親フレーム)」として背景≒index.htmlが表示されていると考えます。
- ・操作対象(出版本)の値を取得しても、それはフレーム①(子フレーム)の値なので、フレーム①に移動しないと使えない
- ・操作対象(出版本)をクリックして表示が変わったフレーム②の値をとりたいが子フレーム間で直接移動は出来ないので一度親フレームに戻る必要がある ①→①→②



① : index.html

```
<html>
<frameset cols="20%,*">
  <frame src="menu.html" name="メニュー"/> ①
  <frame src="main.html" name="メイン"/> ②
</frameset>
</html>
```

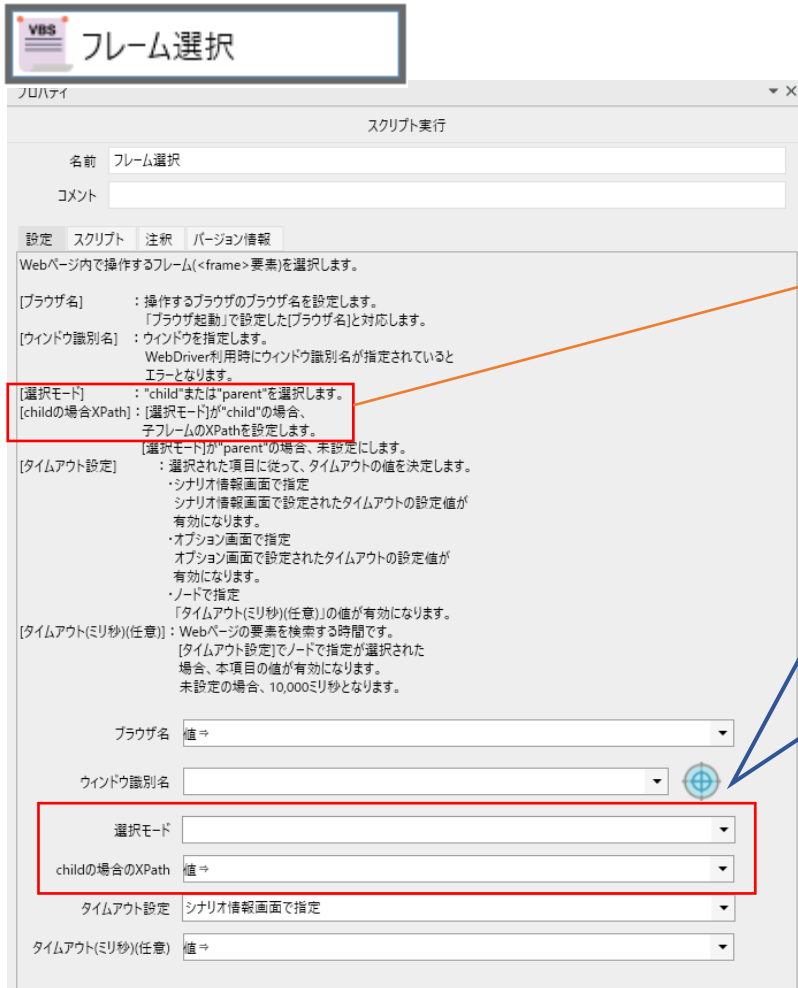
■ ①→① 親フレームから子フレームへ移動
(1) child: **//frame[@name="menu"]**

■ ①→①→② 親フレーム経由で子フレーム移動
(1) parent
(2) child: **//frame[@name="main"]**

「23_ブラウザ関連/フレーム選択」ライブラリの使い方

必要な設定は「**選択モード**(child or parent)」と「**childの場合のXPath**」だけです。

※「**選択モード**」を「parent」にしたときは「**childの場合のXPath**」は入力不要



[**選択モード**] : "child"または"parent"を選択します。
[**childの場合XPath**] : [選択モード]が"child"の場合、
子フレームのXPathを設定します。
[選択モード]が"parent"の場合、未設定にします。

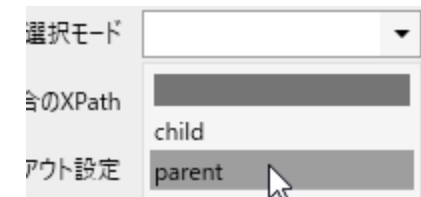
■ ①→①→② 親フレーム経由で子フレーム移動

①→① (親フレームに移動)

(1) 選択モードで「Parent」を選択 だけ

①→②

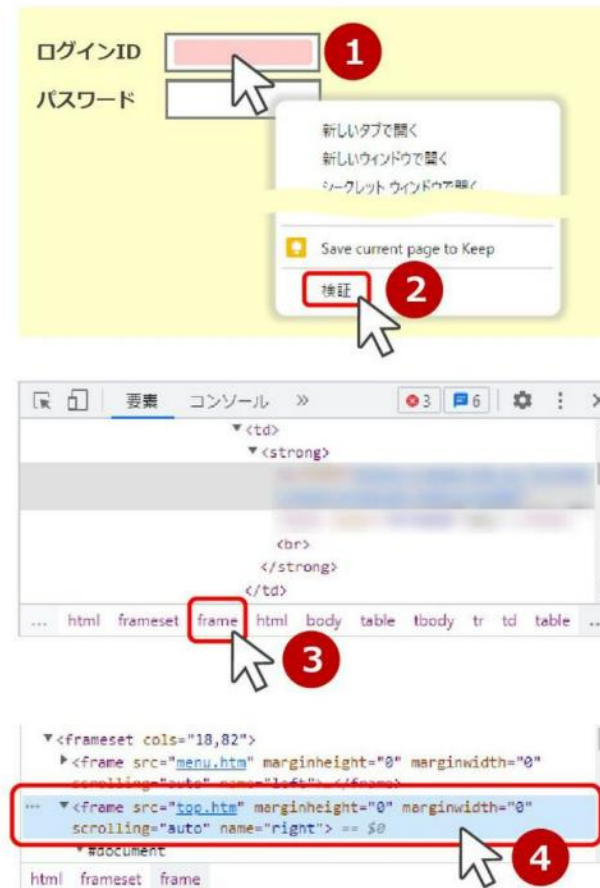
(2) 選択モードで「child」を選択し、
childの場合のXPathに移動先フレームのXPath値を入力: **`://frame[@name="main"]`**



▼ ポイントは③「frame要素をクリック」！ 右にhtmlやbodyが連なっている場合はframeを指すXPathではありません

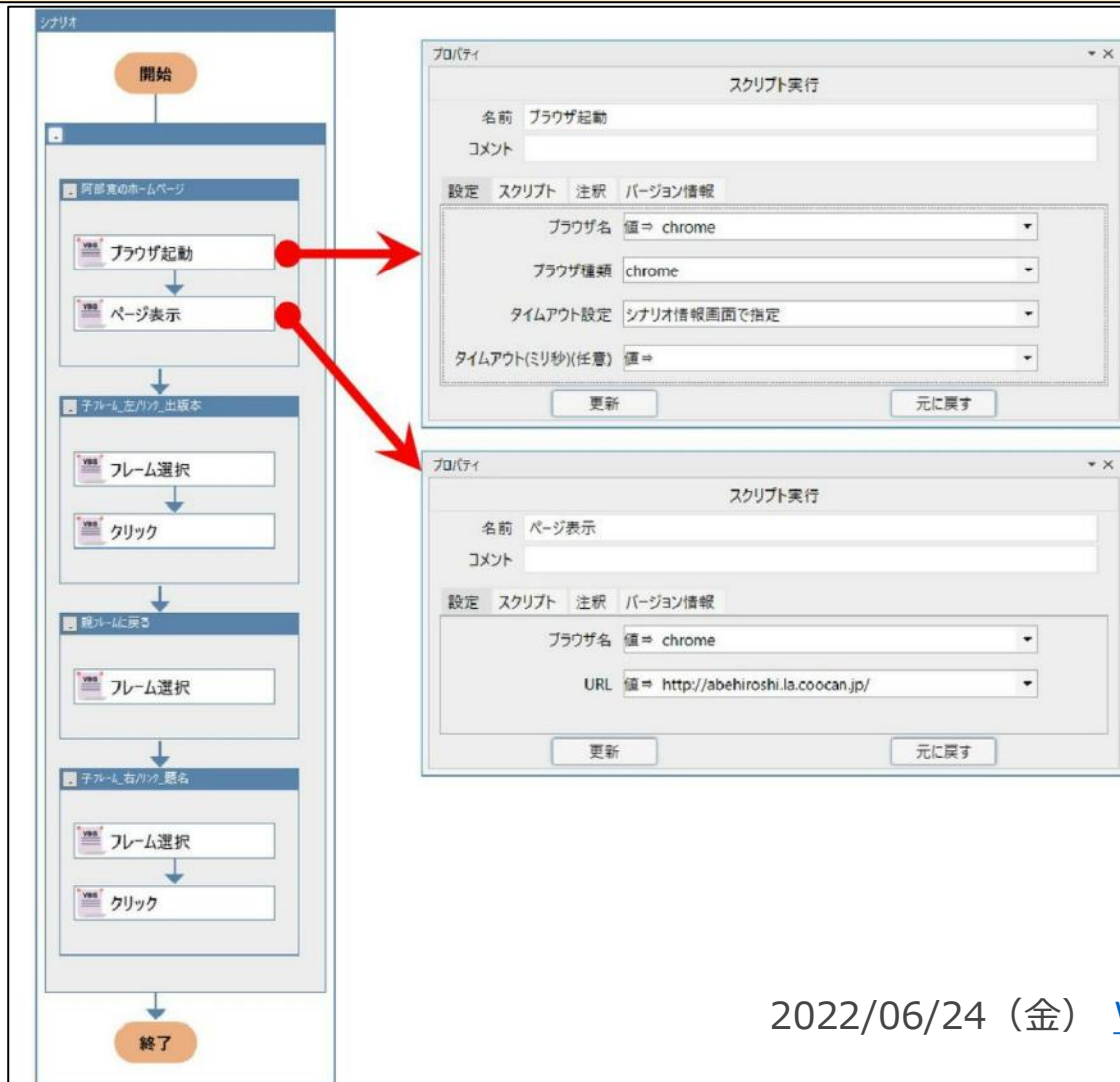
ブラウザの操作ができない場合は……

- ① 調べたい要素の上で右クリック
- ② 表示されたメニューの【検証】をクリック
- ③ HTML階層情報にframe要素があればクリック
- ④ 青くなった箇所を右クリックしてXPathを取得
- ⑤ WinActorライブラリ【フレーム選択】を設定



2022/06/24 (金) [WinActorテクニック披露！橋本名人シリーズvol.3～20時だよ！全員集合～](https://rpacomcommunity.connpass.com/event/250617/presentation/)
<https://rpacomcommunity.connpass.com/event/250617/presentation/>

▼まずはお約束の「ブラウザ起動」、「ページ表示」。この時点では親フレーム(parent)を認識してます。



2022/06/24 (金) [WinActorテクニック披露！橋本名人シリーズvol.3～20時だヨ！全員集合～](https://rpacommunity.comnpass.com/event/250617/presentation/)
<https://rpacommunity.comnpass.com/event/250617/presentation/>

▼「フレーム選択」で左の子フレームframe[1]へ移動。これでようやく「●出版本」がクリック出来ます。

The diagram shows a WinActor scenario with steps: 開始, 阿部寛のホームページ (ブラウザ起動, ページ表示), 子フレーム左の「出版本」 (フレーム選択, クリック), 親フレームに戻る (フレーム選択), 子フレーム右の「映画」 (フレーム選択, クリック), 終了.

Two screenshots of the WinActor 'プロパティ' (Properties) dialog are shown:

- The top screenshot, titled 'フレーム選択', has '名前' (Name) set to 'フレーム選択' and 'childの場合のXPath' (XPath for child) set to '/html/frameset/frame[1]'. It is highlighted with a red dashed border.
- The bottom screenshot, titled 'クリック', has '名前' (Name) set to 'クリック' and 'XPath' set to '/html/body/table/tbody/tr[12]/td[3]/p/a'. It is highlighted with a red solid border.

Step 2: 左側のメニューの「出版本」のリンクをクリック (Click the '出版本' link in the left menu).

フレーム選択の値⇒/html/frameset/frame[1]
XPathの値 ⇒/html/body/table/tbody/tr[12]/td[3]/p/a

2022/06/24 (金) [WinActorテクニック披露！橋本名人シリーズvol.3～20時だヨ！全員集合～](https://rpacommunity.comnpass.com/event/250617/presentation/)
<https://rpacommunity.comnpass.com/event/250617/presentation/>

▼ 右フレームの書籍をクリックしたいが、ポイントは**親フレーム(parent)に戻る処理を間に挟む**こと。
左の子フレームframe[1]から右の子フレームframe[2]にダイレクトに移動はできません！

The diagram shows a WinActor scenario with three frames. The first frame (left) contains 'ブラウザ起動' and 'ページ表示'. The second frame (middle) contains 'フレーム選択' and 'クリック'. The third frame (right) contains 'フレーム選択' and 'クリック'. A red arrow points from the 'フレーム選択' step in the second frame to the 'フレーム選択' step in the third frame. A 'プロパティ' dialog box is open, showing the 'スクリプト実行' tab. The '名前' field is 'フレーム選択', 'コメント' is empty, 'ブラウザ名' is 'chrome', '選択モード' is 'parent', 'childの場合のXPath' is empty, 'タイムアウト設定' is 'シナリオ情報画面で指定', and 'タイムアウト(ミリ秒)(任意)' is empty. A red dashed box highlights the 'フレーム選択' step in the scenario and the 'スクリプト実行' dialog.

ポイント

- 選択モード：childでフレームを選択したら
- 選択モード：parentのフレーム選択で戻り
- 選択モード：childで次のフレームを選ぶ

The screenshot shows a web browser window with a list of books. The list includes 'なぜ最善を尽くさないのか', '映画出演', '出版本', and 'トップ'. A red dashed box highlights the list of books.

2022/06/24 (金) [WinActorテクニック披露！橋本名人シリーズvol.3～20時だヨ！全員集合～](https://rpacomcommunity.com/pass.com/event/250617/presentation/)

<https://rpacomcommunity.com/pass.com/event/250617/presentation/>

▼「フレーム選択」で右の子フレームframe[2]へ移動。これで右フレーム内の要素が選べます。

The diagram shows a WinActor scenario with steps: 開始, 可部見のホームページ (ブラウザ起動, ページ表示), 子フレーム左/右/中央/出稿本 (フレーム選択, クリック), 親フレームに戻る (フレーム選択), 子フレーム右/左/中央/題名 (フレーム選択, クリック), 終了.

Two screenshots of the WinActor 'Script Execution' dialog are shown. The top one is for 'フレーム選択' with XPath `/html/frameset/frame[2]`. The bottom one is for 'クリック' with XPath `/html/body/table/tbody/tr[1]/td/a`.

3 右側のメイン画面に表示される「なぜ最善を尽くさないのか」のリンクをクリック

フレーム選択の値⇒ `/html/frameset/frame[2]`
XPathの値 ⇒ `/html/body/table/tbody/tr[1]/td/a/strong`

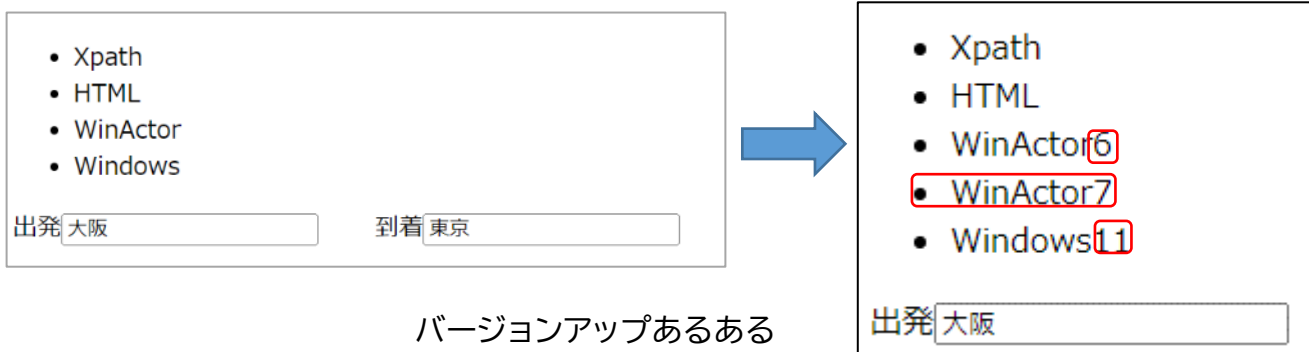
2022/06/24 (金) [WinActorテクニック披露！橋本名人シリーズvol.3～20時だヨ！全員集合～](https://rpacomcommunity.comnpass.com/event/250617/presentation/)
<https://rpacomcommunity.comnpass.com/event/250617/presentation/>

■4章:XPathで値を取っておけば安心?(より変化に強いXPath活用)

4章:XPathにしとけば安心?

HTMLの構造がシナリオ作成時と実行時で異なっていると、正しく要素を指定することができないことがあります。

要素を完全一致で指定していると柔軟性のないシナリオになりかねません。



```
<html>
<body>
<h1>SAMPLE Document</h1>
<div class="main">
<ul>
<li><h1>Test</h1>
<li class="title" id="XPath">XPath</li>
<li class="title" id="HTML">HTML</li>
<li class="title" id="WinActor6">WinActor6</li>
<li class="title" id="WinActor7">WinActor7</li>
<li class="title" id="Windows11">Windows11</li>
</ul>
</div>
出発<input type="text" id="query_input" name="from" value="大阪">
到着<input type="text" id="query_input" name="to" value="東京">
<BR><BR><BR>
<input type="button" id="button1" onclick="alert('Test');" value="ボタン">
</body>
</html>
```

赤字が更新された箇所

対策① XPath関数を利用

開発者ツールで取得した値（完全一致）にXPath関数を適用することで部分一致、n番目指定、and / or / not と言った柔軟性のある指定が可能になります。

例：上から2番めの「WinActor○」、一番下の「WinActor○」

対策② id属性以外でのXPath指定(上級者向け)

過去資料のご紹介とさせていただきます m(_)_m

開発者モードが識別できる属性は「id」のみ。必要に応じ適した属性を使い分ける。

XMLやHTMLには属性(要素内に記述され、要素に性質を与える仕組み)が存在しており、属性を指定することで対象の要素を指定できます

基本構文 : **要素名[@属性名=属性の値]**で記述する。

```
<li class="title" id="WA">WinActor</li>
```

この例でタグの”id”属性が”WA”で表せるため、id属性を指定して以下のように表記できる
//li[@id="WA"] 取得結果: ”WinActor

関数の構文: **要素名[関数名(@属性名, ”関数が扱う文字列”)]** で記述する。
括弧が2種類になったこと、イコールじゃ無くなった事に注意

条件指定の構文: **要素名[条件A 条件指定 条件B]**で記述する。
条件指定にはand、or、notが使える。
また「and not」⇒「～かつ」「～が～ではない」のような同時利用も可能。
各条件は前述の基本構文または関数構文で記述する。

XPath関数(部分一致、前方一致、位置指定)

■「タグの属性に対し、部分一致や前方一致などで要素を指定できます。」

① **contains関数** : **部分一致**。特定の文字列が含まれる要素を指定する

構文: contains(@属性名, "属性値に含まれる文字列")

例: "li"タグの"id"属性に"path"を含む要素を指定

//li[contains(@id,"path")] → 取得結果: XPath

② **starts-with関数** : **前方一致**。特定の文字列から始まる要素を指定する

構文: starts-with(@属性名, "属性値を前方一致検索する文字列")

例: "li"タグの"id"属性が"Win"で始まる要素を指定

//li[starts-with(@id,"Win")] → 取得結果: WinActor

③ **position関数** : **要素の位置を指定**する

構文: position() = "要素の番号" ※ li[]で代用可。

例: "li"タグの2番目の要素を指定

//li[position() = 2] → 取得結果: HTML

最後の要素は「last()」で指定できます。

```
<html>
<body>
<div class="main">
  <ul>
    <h1>Test</h1>
    <li class="title" id="XPath">XPath</li>
    <li class="title" id="HTML">HTML</li>
    <li class="title" id="WinActor">WinActor</li>
  </ul>
</div>
</body>
</html>
```

XPathで条件指定(AND,OR,NOT)

前述の構文: **要素名[条件A 条件指定 条件B]**で記述する。

①AND条件

右例で”li”タグの「idが”WinActor”かつテキストに”7”を含む」値を取得

```
//li[@id="WinActor" and contains(text(),"7")]
```

→取得結果:WinActor7

```
<li class="title" id="XPath">XPath</li>
<li class="title" id="HTML">HTML</li>
<li class="title" id="WinActor">WinActor</li>
<li class="title" id="WinActor">WinActor7</li>
```

②OR条件

右例で”li”タグの「idが”WinActor”か”H”を含む」データの2番目を取得

```
//li[contains(@id,"WinActor") or contains(@id,"H")][2]
```

→取得結果:WinActor

```
<li class="title" id="XPath">XPath</li>
<li class="title" id="HTML">HTML</li>
<li class="title" id="WinActor">WinActor</li>
<li class="title" id="WinActor">WinActor7</li>
```

③NOT条件

右例で”li”タグの「idが”WinActor”ではない」データの1番目を取得

```
//li[not(@id="WinActor")]
```

→取得結果:XPath

```
<li class="title" id="XPath">XPath</li>
<li class="title" id="HTML">HTML</li>
<li class="title" id="WinActor">WinActor</li>
<li class="title" id="WinActor">WinActor7</li>
```

■XPath関数条件指定記述についての練習問題

問題①「到着」テキストボックスの値を指定する

ただし”id”を用い”query-input”の2つ目を指定する

開発者ツールで取得 `://*[@id="query_input"]`

WinActorで取得 `://*[@name='to']`

属性が異なります。
「XPathをコピー」が
識別するのはId属性だけ

```
<html>
<body>
  <h1>SAMPLE Document</h1>
  <div class="main">
    <ul>
      <h1>Test</h1>
      <li class="title" id="XPath">XPath</li>
      <li class="title" id="HTML">HTML</li>
      <li class="title" id="WinActor">WinActor</li>
      <li class="title" id="Windows">Windows</li>
    </ul>
  </div>
  出発<input type="text" id="query_input" name="from" value="大阪">
  到着<input type="text" id="query_input" name="to" value="東京">
  <BR><BR>
  <input type="button" id="button1" onclick="alert('Test');" value="ボタン">
</body>
</html>
```

同じidが
2つあります！

解答①

～の**2つ目を指定**＝「position」関数を使用

`://*[@id="query_input"][position() = 2]`

SAMPLE Document

Test

- Xpath
- HTML
- WinActor
- Windows

出発 到着

■XPath関数・条件指定記述についての練習問題

問題②「WinActor」の値を取得する

ただし、「id」が「WinA」から始まるものとする

開発者ツールで取得 `://*[@id="WinActor"]`

WinActorで取得 `:この例では対象の要素は`

`「XPath選択」ボタンから取得不可。(指定は可能)`

解答②

～から始まる = 「starts-with」関数を使用

`://*[starts-with(@id, "WinA")]`

```
<html>
<body>
  <h1>SAMPLE Document</h1>
  <div class="main">
    <ul>
      <h1>Test</h1>
      <li class="title" id="XPath">XPath</li>
      <li class="title" id="HTML">HTML</li>
      <li class="title" id="WinActor">WinActor</li>
      <li class="title" id="Windows">Windows</li>
    </ul>
  </div>
  出発<input type="text" id="query_input" name="from" value="大阪">
  到着<input type="text" id="query_input" name="to" value="東京">
  <BR><BR>
  <input type="button" id="button1" onclick="alert('Test');" value="ボタン">
</body>
</html>
```

SAMPLE Document

Test

- Xpath
- HTML
- WinActor
- Windows

出発

到着

■XPath関数・条件指定記述についての練習問題

問題③id指定で「出発」テキストボックスの値を取得する

ただし "id"が"query-input"かつ "name"が"to"ではない

//*[@id="query_input"] かつ

//*[@name='to'] ではない

解答③

「~**かつ**」「 ~が~**ではない**」(「and」と「not」の同時利用)

//*[@id="query_input" **and not**(@name='to')]

```
<html>
<body>
  <h1>SAMPLE Document</h1>
  <div class="main">
    <ul>
      <h1>Test</h1>
      <li class="title" id="XPath">XPath</li>
      <li class="title" id="HTML">HTML</li>
      <li class="title" id="WinActor">WinActor</li>
      <li class="title" id="Windows">Windows</li>
    </ul>
  </div>
  出発<input type="text" id="query_input" name="from" value="大阪">
  到着<input type="text" id="query_input" name="to" value="東京">
  <BR><BR><BR>
  <input type="button" id="button1" onclick="alert('Test');" value="ボタン">
</body>
</html>
```

SAMPLE Document

Test

- Xpath
- HTML
- WinActor
- Windows

出発

到着

開発経験ゼロでもここまではすぐ出来るようになります！

この入門編を入り口とし、RPACommunityで公開されてる高度な対応例に挑戦してください。

1例として 2022/04/25 WinActor Talk 「Chromeブラウザでのシナリオ作成を徹底解説」

<https://rpacommunity.connpass.com/event/244762/>

開発ツールから取得されるXPathの問題点

● 開発ツールでは、現在ブラウザが表示している要素を指定するXPathが取得されますが、シナリオ実行時に、お知らせのメッセージが増えている、検索結果のページが異なっていたり、HTMLの構造がシナリオ作成時とシナリオ実行時で異なっていると、そのままでは正しい要素を指定することができないことがあります。

昨日まで5ページしかなかったのに、今日は6ページあり、次ページのリンクをクリックできず、シナリオの実行に失敗した！

■ 昨日（シナリオ作成時）のXPath
`/html/body/div[2]/div[1]/div/div/a[6]`

■ 今日（シナリオ実行時）のXPath
`/html/body/div[2]/div[1]/div/div/a[7]`

Webページの構造変化に強いXPathとは

● Webページの構造の変化に強いXPathを選択します。

■ 「XPathをコピー」※1から得られたXPath
`//*[@id="js_fs_paginate"]/div/a[6]`

■ 「完全なXPathをコピー」※2から得られたXPath
`/html/body/div[2]/div[1]/div/div/a[6]`

左に比べこの部分の変化に弱い

id属性にて、ページ内の要素を特定することができるため、操作したい対象の要素に近いid属性で特定できる要素を選択し、XPathで利用します。ブラウザの開発ツールの「XPathをコピー」では、この選択は自動的に行ってもらえるので、そのまま利用できます。id属性で特定できる要素が無い場合は、「XPathをコピー」と「完全なXPathをコピー」で同じXPathが取得されます。

id属性で指定される要素より親側の变化に強くなったが、前のページの子供側の变化に対応できていない！

※1 Edgeでは「XPathをコピー」、Chromeでは「Copy XPath」、Firefoxでは該当メニューなし
※2 Edgeでは「完全なXPathをコピー」、Chromeでは「Copy full XPath」、Firefoxでは「XPath」

構造変化に強いXPathとは（その2）

● id属性以外の方法による要素の特定方法について、いくつか紹介します。

■ a要素内のテキストが「次へ」である場合（`次へ`）
`//*[@id="js_fs_paginate"]/div/a[text()='次へ']`
[text()='文字列']にて、要素の内部の文字列が一致する要素が選択されます。

■ a要素内のテキストが「次へ」を含む場合（一致はしない場合）（`>次へ`）
`//*[@id="js_fs_paginate"]/div/a[contains(text(),'次へ')]`
[contains(text(),'文字列')]にて、要素の内部の文字列が部分一致する要素が選択されます。[contains(@class,'label')]といった使い方もできます。

■ 最後に現れるa要素の場合
`//*[@id="js_fs_paginate"]/div/a[last()]`
[last()]にて、最後の要素を取得することができます。[0]で先頭の要素

ブラウザの開発ツールから取得されるXPathについて、id属性にて特定できない要素に対しては、テキスト、他の属性、及び、順番といった様々な条件（XPathでは「」に記述したもの以外に多くの条件を用いることができます）を組み合わせていくことで、変化に強いXPathを設定できます。23 ブラウザ関連/ツール/スクレイピングライブラリの、ページ選択要素XPathの選択ボタンではテキストによる条件を用いたXPathを自動的に作成していますので、参考にしてください。

IE操作の対象コントロール指定をXPathで行うには？

● IE操作では対象コントロールの指定にて、XPathではなくtag、frame index、tag index等を利用しています。
● IEのサポート終了に伴い、Edge等の別のブラウザによる操作へ置き換える方法はないか？

■ 「対象コントロール指定」で得られる詳細設定

変化は弱い、現シナリオが安定しているならば試したい。ブラウザの種類により表示を変えるページもあり、必ずしも一致はしません。

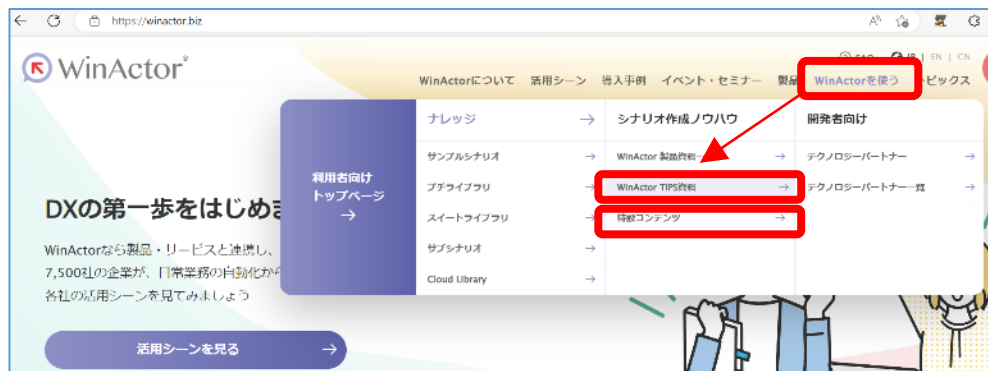
現時点では、動作確認が必要なものとなりますが、以下のXPathで概ね代用できますので、お試しください。

`//*[@tag=<tag index>][@name=<name>][@type=<type>][@id=<id>][@value=<value>]`

<frame index>が0以外の場合、23 ブラウザ関連/フレーム選択ライブラリを用いたフレームの選択が必要です。

今後もコンテンツを拡充していきます！

入門、応用、コンセプト紹介などまとめて掲載してます。ぜひご覧ください



WinActorを使う > WinActor Tips資料(pdf)

- ・変化に強いXPathの指定のコツ
- ・WinActorライセンスの説明資料
- ・Ver.6をご利用のお客様向けのWinActor Ver.7移行参考資料

WinActorを使う > 特設コンテンツ > 特集動画

初出:RPACommunity

- 【WinActor】エラーが出て怖くない！シナリオが止まった時のヒントを紹介！
- 【NTT-AT直伝】WinActorとAPI連携を学ぶ
- WinActor Ver.7への移行、分からないことはこの動画で解決！
- 最新！WinActor Ver.7.4 開発ベンダー(NTT-AT)から最速で説明
- 【WinActor】IEサポート終了に伴う Edge IEモードへの移行対応
- 【2021秋】WinActor Ver.7.3 の情報を紹介！NTTアドバンステクノロジー登壇
- Microsoft365やGoogle Workspaceと連携もできる！管理ツールの紹介も有！
- 【爆速】遂にWinActor ver7.2が出た！気になる新機能をこの動画で一挙にご紹介！【イベント未公開分】
- ようこそ！WinActor Ver.7の世界へ！

NTT-AT 純正

- WinActorシナリオ作成動画:ブラウザを起動後ログインページのIDとパスワードを自動入力
- WinActorでエアコン取付工事指示書を作成(音声なし)
- WinActorで実現しよう 全社DX ~full Ver.~【コンセプト紹介】
- WinActorで実現しよう 全社DX ~Short Ver.~【コンセプト紹介】
- かんたん導入 WinActor ~Short Ver.~【コンセプト紹介】

「WinActorと協働し
誰もがクリエイティブに働ける社会へ」

ご清聴ありがとうございました

WinActorメーカー
公式ホームページ
<https://winactor.biz>

未来を拓くチカラと技術。

