XPath学習資料

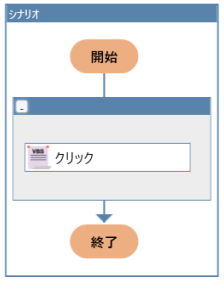
NTTアドバンステクノロジ株式会社

XPathを使うメリット

XPathによりWebページの部品（HTMLの特定箇所）を指定することができる。

画像識別方式とは異なり、画面の表示状態に影響されないことから

安定したブラウザ操作シナリオを作成することが可能。



「6」ページ目に移動するボタンは
XPathでは下記の通り表現される。

//*[@id="js_fs_paginate"]/div/a[6]

WinActorはXPathを基にwebページを操作可能で
左図のように設定を行うことで
正確にボタンをクリックする操作を実装することができる。

Webページの構造変化に対応するには

XPathを活用してもWebページの構造変化には影響を受けてしまう。

例えば、「 ＞次のページへ進む 」ボタンをクリックしたいという操作があった場合に

//*[@id="js_fs_paginate"]/div/a[6]

//*[@id="js_fs_paginate"]/div/a[7]

「昨日まで全5ページだったが、今日は6ページあり、ボタンをクリックできなかった。」ということが発生する。

つまりボタンが6番目の要素であったものが、構造変化により7番目にズレてしまったということになる。

XPathの”関数”を活用することでこのようなケースに対応できるようになる。

本資料ではこうした関数の使用方法を一部紹介。

このケースでは、「 ＞次のページへ進む 」ボタンが要素の最後に設置されるという規則性があることから

要素の最終行をクリックするように設定変更するとページ追加によるエラーを回避することができる。

//*[@id="js_fs_paginate"]/div/a[position() = last()]

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

```
<html>
 <body>
  <h1>Sample Document</h1>
  <div class="main">
   <ul>
    <li class="title">Xpath</li>
    <li class="title">HTML</li>
    <li class="title">WinActor</li>
   </ul>
  </div>
 </body>
</html>
```

左記HTMLから”Sample Document”の文字を取得するXPath

/html/body/h1

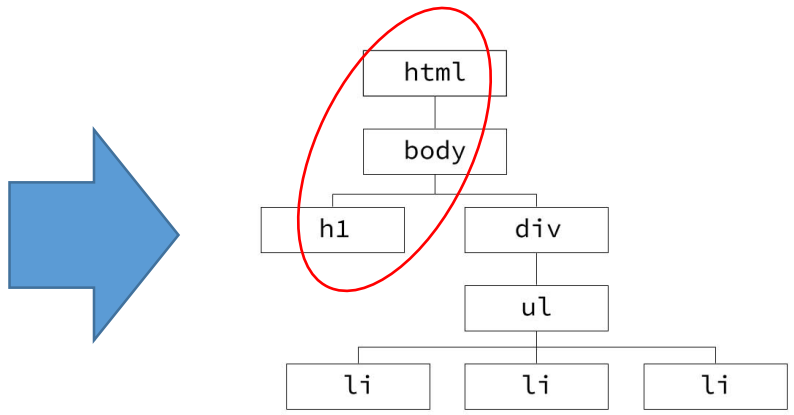
XPathの記述方法①

①タグ(要素)の階層構造で記述

XMLやHTMLは階層構造(ツリー構造)となっているため、その特性を利用して対象の要素を指定する方法

HTMLの階層構造を利用し、各ノード(節点)を「/(スラッシュ)」で区切り、要素までの道のりを示す

```
<html>
  <body>
    <h1>Sample Document</h1>
    <div class="main">
      <ul>
        <li class="title">Xpath</li>
        <li class="title">HTML</li>
        <li class="title">WinActor</li>
      </ul>
    </div>
  </body>
</html>
```



上記例で"Sample Document"の文字列には<html><body><h1>のノードを辿ることで

到達できるため、それぞれのノードを"/"で区切ると以下のように表記できる

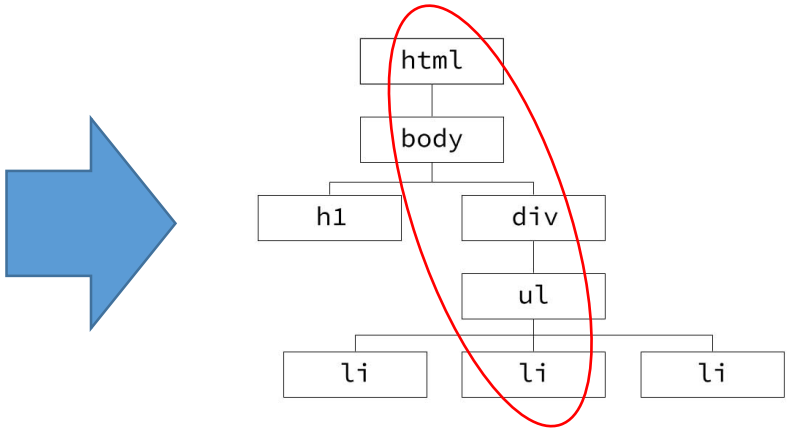
/html/body/h1

XPathの記述方法②

■同じ階層に同じ要素が複数存在する場合の表記

同じ階層に同じ要素が複数存在する場合、何番目の要素を取り出すか[]の中に番号を指定して表す

```
<html>
  <body>
    <h1>Sample Document</h1>
    <div class=”main”>
      <ul>
        <li class=”title”>Xpath</li>
        <li class=”title”>HTML</li>
        <li class=”title”>WinActor</li>
      </ul>
    </div>
  </body>
</html>
```



上記例で”HTML”の文字列は<html><body> の階層にあるが、同階層に同じ<li>要素が3つ存在するため、2番目の<li>要素であることを表すよう以下のように記述する

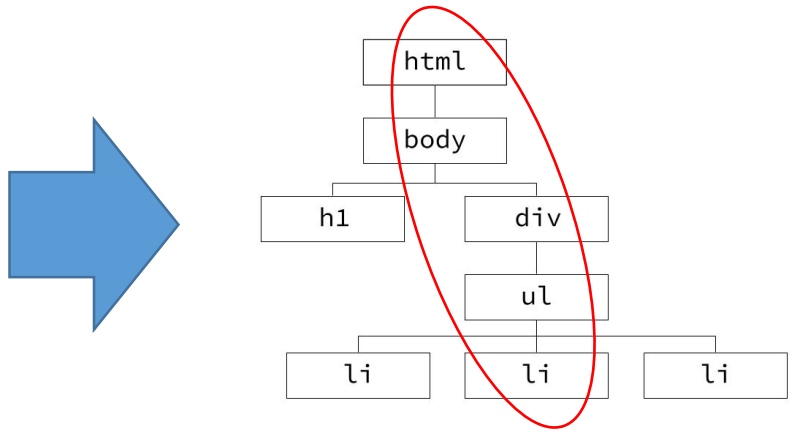
/html/body/div/ul/li[2]

XPathの記述方法③

■XPathの省略

毎回ルートからパスを記述するとXPathの記述が長くなり見づらくなるが、“//”を用いて途中までのパスを省略して記述することができる

```
<html>
  <body>
    <h1>Sample Document</h1>
    <div class=”main”>
      <ul>
        <li class=”title”>Xpath</li>
        <li class=”title”>HTML</li>
        <li class=”title”>WinActor</li>
      </ul>
    </div>
  </body>
</html>
```



ルートから要素を辿った表記(絶対XPath):/html/body/div/ul/li[2]

までのパスを省略した表記://ul/li[2]

途中のパスをすべて省略した表記://li[2]

XPathの記述方法④

■XPath省略時の注意点

省略したXPathが表す要素が複数存在した場合、指定したい要素以外が対象になることがあるため要素が一つになるよう絞り込んで指定する

```
<html>
 <body>
  <h1>Sample Document</h1>
  <div class="main">
   <ul>
    <h1>Test</h1>
    <li class="title">Xpath</li>
    <li class="title">HTML</li>
    <li class="title">WinActor</li>
   </ul>
  </div>
 </body>
</html>
```

絶対XPath：/html/body/div/ul/h1

→取得結果：Test

途中のパスをすべて省略した表記：//h1

→取得結果：Sample Document

上記例で"Test"を取得する場合、途中のパスをすべて省略するとXPathが示す要素が

「<h1>Sample Document</h1>」と「<h1>Test</h1>」の複数存在することになり

正しい結果を得ることができないため、直前の<ul>は省略せずに以下のように記述する

//ul/h1

XPathの記述方法⑤

②属性の指定による記述

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する記述方法

属性を「@（アットマーク）」で表し、要素名[@属性名＝属性の値]の構文で記述する

```
<html>
  <body>
<div class=”main”>
     <ul>
       <h1>Test</h1>
       <li class=”title” id=”XP”>Xpath</li>
       <li class=”title” id=”HT”>HTML</li>
       <li class=”title” id=”WA”>WinActor</li>
     </ul>
   </div>
  </body>
</html>
```

上記例では”WinActor”の文字列は<li>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”]

XPathの記述方法⑥ – XPath関数①

■XPath関数による属性の指定

XPathの標準関数を使用することで属性の部分一致検索や要素の位置による指定を行うことができる

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

構文: contains(@属性名,”属性値に含まれる文字列”)

```
<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>
```

上記例で”li“タグの”id”属性に”path”を含む要素を指定するには以下の通り記述する

//li[contains(@id,”path”)]

→取得結果:Xpath

XPathの記述方法⑥ – XPath関数②

■XPath関数による属性の指定

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

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

```
<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>
```

上記例で”li“タグの”id”属性が”Win”で始まる要素を指定するには以下の通り記述する

//li[starts-with(@id,”Win”)]

→取得結果:WinActor

XPathの記述方法⑥ – XPath関数③

■XPath関数による属性の指定

③ position()：要素の位置を指定する

構文： position() = ”要素の番号”

```
<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>
```

上記例で”li“タグの2番目の要素を指定するには以下の通り記述する

//li[position() = 2]

→取得結果：HTML

※ **li[2]と同義のため省略されることが多い**

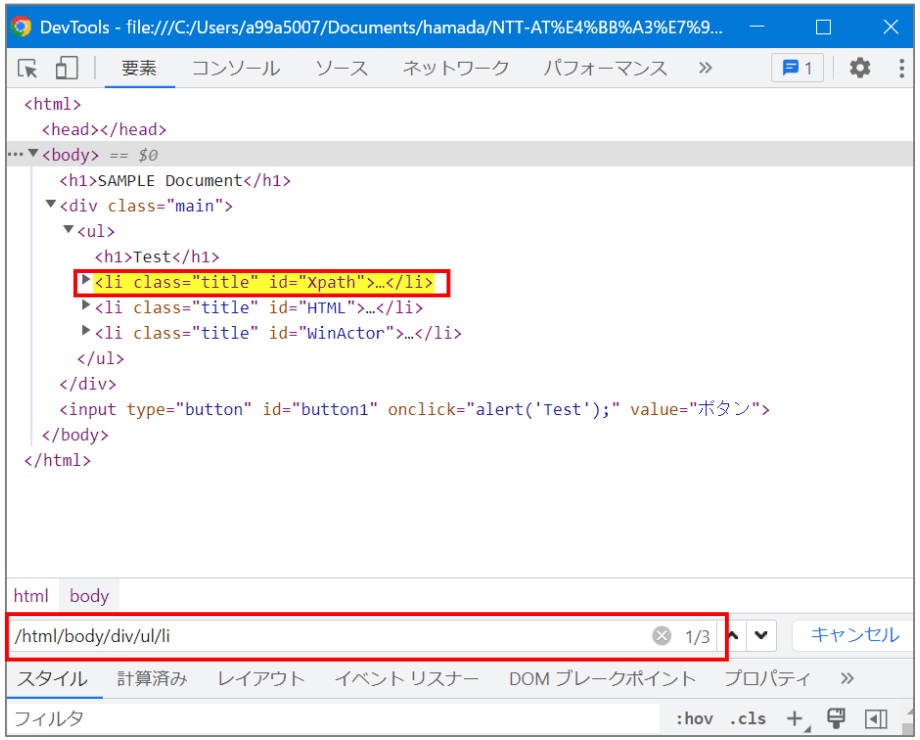
数字の代わりに

最後の要素を指定する場合は「 last() 」が利用可能

開発者ツール使用時のポイント①

■XPathでのコンテンツ検索

・ブラウザの開発者モード使用時に「Ctrl＋F」キーを押して検索窓を立ち上げXPathを入力すると該当する要素と数を検索できるので、XPathの記述が正しいかを簡単に確認できる



【使用方法】

① 「F12」キーを押してブラウザの開発者モードを起動する

② 「要素」タブ上で「Ctrl+F」キーを押し検索窓を表示する

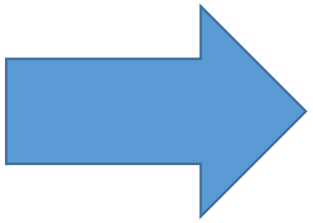
③ 表示された検索窓にXPathを入力するとHTML内の該当する部分が黄色でマークされ検索窓の右側に該当箇所がいくつあるかが表示される

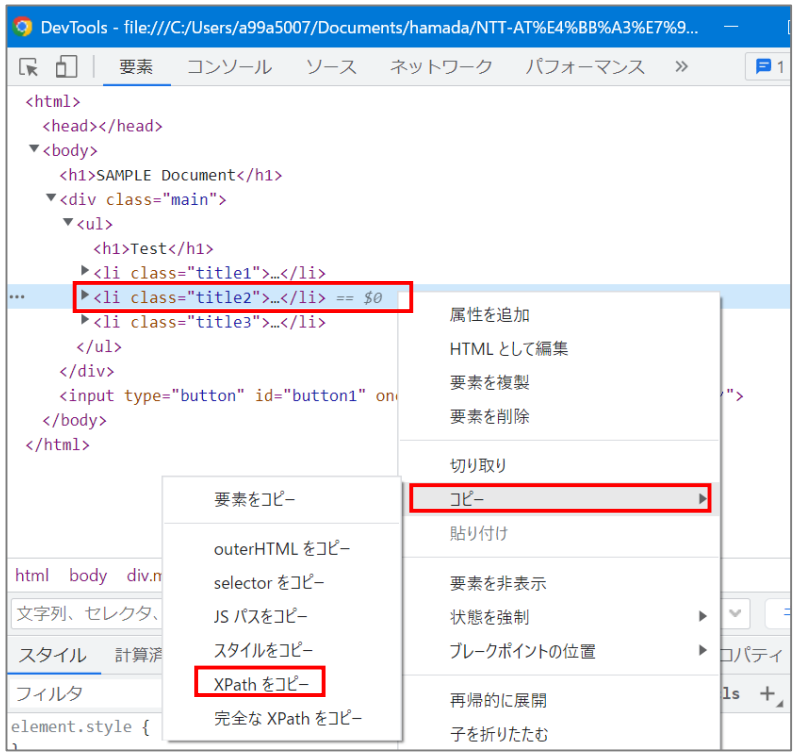
開発者ツール使用時のポイント②

■XPathコピー時の仕様について

・ブラウザの開発者モードで要素を選択して右クリックし「コピー」→「XPathをコピー」でXPathをクリップボードに取得できるが、この際に識別している属性は「id」のみとなるため、「id」以外の項目で要素を特定できる場合でも最適化されたXPathにならないため注意する

```
<html>
  <body>
<div class=”main”>
     <ul>
       <h1>Test</h1>
       <li class=”title1”>Xpath</li>
       <li class=”title2”>HTML</li>
       <li class=”title3”>WinActor</li>
     </ul>
   </div>
  </body>
</html>
```





上記HTMLで「HTML」の文字を取得するため開発者モードでXPathをコピーして取得すると「/html/body/div/ul/li[2]」となる。

本来はclass属性が異なるため省略すると「//li[@class=“title2”]」と記述できるが、開発者ツールではid属性しか識別しないためclass属性は省略されない

参考:よく使用されるHTMLタグ

■HTML中でよく使用されるタグと概要

タグ名	概要
html	HTML文書であるという指定
head	文書のヘッダ部分を指定
title	文書のタイトルを指定
body	文書の本体部分を指定
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
form	入力・送信フォームを指定
input	入力部品を指定
select	セレクトボックスを指定
button	ボタンを指定
h1～h6	見出しを指定
p	段落を指定
div	ブロック要素を指定
table	表を指定
tr	表の行を指定
td	表の列を指定