

5-1

TSUTAYA ONLINEでレンタル作品の
在庫を店舗ごとに検索する

ワークシート上に記入した複数のTSUTAYA店舗について、レンタルCDやDVDの在庫有無を確認することができるサンプルです。本サンプルを利用することで自宅周辺や通勤・通学経路上の「どの店舗に在庫があるか」という、TSUTAYA ONLINEの検索機能とは別の切り口の情報を手間なく調べることができます。



©角川書店

©2010 第501統合戦闘航空団

	実行		
12			
13	対象店舗	在庫有無	
14	自由が丘	[×]	
15	学芸大	[-]	
16	祐天寺	[-]	
17	中日黒	[-]	
18	田町駅前	[-]	
19	六本木ヒルズ	[○]	
20			

[実行] ボタンをクリックすると店舗ごとの在庫検索が始まる

Excel VBAを実行すると商品ページから在庫を確認できる

スプレッドシートを備えるExcel VBAならではのツールを作ろう

「ワークシートに一覧化されたデータをキーにWebページで情報を収集し、その結果をワークシートに書き戻す」これは、スプレッドシートを標準で備えたExcel VBAのいわば「十八番」の処理と言えるでしょう。さて、今回の操作対象となるTSUTAYA ONLINEには、作品の「店舗名を指定して在庫検索」という機能がありますが、在庫のある店舗を一覧化する機能はありません。このサンプルでは、あらかじめワークシートに一覧化した店舗に対して「店舗名を指定して在庫検索」を連続実行することで、各店舗の在庫状況を画面からワークシートに転記するようにしています。

1 自動化の範囲を決める

まずは手動で在庫検索する場合の流れを確認し、自動化の範囲を決めます。

- ① TSUTAYA ONLINEのトップページ（<http://store-tsutaya.tsite.jp/>）を開きます。
- ② 画面上部の検索ボックスにレンタルしたい作品名の一部または全部を入力し、[検索] ボタンをクリックする。
- ③ 作品検索結果一覧画面から対象の作品をクリックし、作品ページを開く。
- ④ 作品ページの [店舗を指定して在庫検索] ボタンをクリックし、店頭在庫検索画面を開く。
- ⑤ 店舗の検索ボックスに店舗名を入力し、[検索する] ボタンをクリックする。
- ⑥ 店舗検索結果一覧画面から対象の店舗の [在庫を表示する] ボタンをクリックし、指定店舗の在庫状況が表示された作品ページを開く。
- ⑦ さらに他の店舗の在庫を調べるためには、[店舗を指定して在庫検索] ボタンをクリックし、再度店頭在庫検索画面を開き、⑤～⑥を繰り返す。

このサンプルでは、④～⑦の処理を自動化します。同じような操作を繰り返し行う部分なので、自動化による手間の削減効果が大きい部分であることがその理由です。

一方、①～③の処理はたいして手間の削減にならない（作品名はWeb/Excelいずれかに結局手入力が必要）だけでなく、③の作品検索結果一覧画面から本当に正しい作品を選択することが運用上困難（シリーズもの等でも正確なタイトルの入力が必要）なため、自動化に適しているとは言えません。

今後オリジナルのツールを作る上でも、闇雲に自動化するのではなく、最も効果のある部分に限って自動化するという観点を意識するとよいでしょう。

お店の在庫状況

よく利用する店舗をマイTSUTAYABOXで「お店登録」されると在庫検索が簡単にできます。

表示されているお店の在庫状況は、最終更新日時時点での情報であり、お問い合わせいただいた時点での店舗在庫状況ではありません。ご利用前には、必ずお店に在庫状況を確認ください。

TSUTAYA 自由が丘店

住所 : 〒152-0035 東京都目黒区自由が丘2-10-8
 電話番号 : 03-5731-6055
 営業時間 : 朝 09:00～深夜02:00

在庫状況 : [X] レンタル返却予定日:2013年3月26日(最終更新日時:2013年3月20日 13時00分頃)

店舗を指定して在庫検索

店舗の在庫状況のWebページを開く操作を自動化する

2 ユーティリティ処理の作成

自動化の範囲を決めたことで、必要となる機能がいくつか見えてきました。具体的には、手動で行うP.147 ①の①の処理で開いたWebページを操作するため、既に開いているIEウィンドウをオブジェクトとして取得する機能が必要になります。また、P.147 ①の④～⑦は画面移動を伴うため、画面移動の完了待ち機能が必要になります。

これらの機能はツールのメイン処理の中に直接記述しても構いませんが、ツールの処理の見通しが悪くなるだけでなく、今後他のツールを作るときに再利用できる方がよいので、ここではそれぞれ独立したプロシージャとして実装することになります。

また、コードを書き始める前は、参照設定で「Microsoft Internet Controls」と「Microsoft HTML Object Library」を追加します。参照設定の手順については、「2-2 データ型を明示して入力補完を活用する」の説明を参照してください。

参照 2-2

```
1 Private Function getIE(arg_title As String) As InternetExplorer ← タイトルを指定してIEを取得
2     Dim ie As InternetExplorer
3     Dim sh As Object
4     Dim win As Object
5     Dim document_title As String
6     Set sh = CreateObject("Shell.Application")
7     For Each win In sh.Windows
8         document_title = ""
9         On Error Resume Next
10        document_title = win.document.Title
11        On Error GoTo 0
12        If InStr(document_title, arg_title) > 0 Then ← タイトルバーに引数が含まれるかチェック
13            Set ie = win
14            Exit For
15        End If
16    Next
17    Set getIE = ie
18 End Function
19
20 Private Sub waitBrowsing(ie As InternetExplorer) ← 画面移動の完了待ち
21     Do While ie.Busy Or ie.ReadyState < READYSTATE_COMPLETE ← IEがBusyまたはReadyStateがCOMPLETEになっていない間はループし続けて先に進まない
22         DoEvents
23     Loop
24 End Sub
```

① InternetExplorerオブジェクト取得プロシージャ。String型の引数「arg_title」は、取得したいIEウィンドウのタイトルです。これを含むIEを検索し、InternetExplorer型のオブジェクトとして返却します。

② Webページ移動完了待ちプロシージャ。InternetExplorer型の引数「ie」は、移動完了の待ち受け処理を行う対象のIEウィンドウです。

①の処理内容はShell.Applicationのオブジェクトに含まれるウィンドウを順次評価する方法です。説明の詳細は「3-4 起動済みのIEを制御する」を参照してください。 **参照 3-4**

②では、BusyプロパティとReadyStateプロパティの状態監視を行い、操作可能な状態になったらループを脱出して呼び出し元に処理が戻ります。説明の詳細は「3-3 画面移動の完了を待つ」を参照してください。 **参照 3-3**

3 作品ページから店頭在庫検索画面に移動する

初めにTSUTAYA ONLINEのWebページから適当なレンタル作品のページを開いてください。ここでは例として「http://store-tsutaya.tsite.jp/item/rental_dvd/089704291.html」を開きます。(P.147 ①の①～③の手順)

まずはHTMLソースから、店頭在庫検索画面に移動するためにクリックするボタンが記述されている箇所を探しましょう。ページ上で右クリックして「ソースの表示」を選択し、表示されたHTMLソース中から「店舗を指定して在庫検索」というキーワードで検索します。

```
<a href="http://store-tsutaya.tsite.jp/tsutaya/top?account=tsutaya&acmd=1&arg=http%3A%2F%2Fstore-tsutaya.tsite.jp%2Fitem%2Frental_dvd%2F089704291.html"></a>
```

ボタンは、aタグ（ハイパーリンク）の設定されたimgタグ（画像）であることがわかりました。aタグ・imgタグのいずれのclickメソッドを実行しても画面の移動は行われませんが、今回はimgタグをクリックします。aタグにはhref属性（移動先URL）しか定義されておらず、またこの値は作品に応じて変化するため、クリック対象として特定するのが困難なためです。

```

1 Private Sub getStock()
2     Dim ie As InternetExplorer
3     Set ie = getIE("TSUTAYA") ← TSUTAYAの画面を取得
4     Dim htdoc As HTMLDocument
5     Set htdoc = ie.document ← ドキュメントを取得
6
7     Dim img As HTMLImg ← 画像評価用のオブジェクト変数
8
9     For Each img In htdoc.getElementsByTagName("IMG")
10        If InStr(img.alt, "店舗を指定して在庫検索") > 0 Then
11            img.Click ← 条件に合致した画像をクリック
12        End If
13    End For
14 Next
15 End Sub

```

- ① 手動で開いた作品画面を取得します。
- ② 画面に含まれるすべてのimgタグ(画像)を対象に処理を実行します。
- ③ alt属性(画像が表示されない場合の代替文字列)が「店舗を指定して在庫検索」の場合、クリックしてループを抜けます。

title属性(ツールチップ表示の補足情報)やsrc属性(画像ファイルのパス)で特定することも可能で、その場合はそれぞれIF文の条件を下記のようにしましょう。

処理を実行して、店頭在庫検索画面に移動できることを確認してください。

```

10 If InStr(img.title, "店舗を指定して在庫検索") > 0 Then ← title属性で特定する場合

```

```

10 If InStr(img.src, "btn_store-fix-stock-search.png") > 0 Then ← src属性で特定する場合

```

4 店舗名を入力して店舗一覧を表示する

店頭在庫検索画面では、店舗名を入力して「検索」ボタンをクリックする処理を行います。

ソースを表示して「店舗名 or 住所の一部を入力」を検索することで店舗名入力テキストボックスを、「検索する」を検索することで「検索」ボタンを、それぞれ見つけることができます。

```
<input name="SearchKey1" type="text" value="店舗名 or 住所の一部を入力" onkeydown="KeyDownSearchKey1(event);"
class="search_word_input" style="COLOR: #999;"
onblur="if (this.value == '') this.value = '店舗名 or 住所の一部を入力';if (this.value == '店舗名 or 住所の一部を入力') this.style.color = '#999';"
onfocus="if (this.value == '店舗名 or 住所の一部を入力') this.value = '';this.style.color = '#000';" />
```

```
<input name="image" type="image" class="toICstCondSearchBtn" title="検索する" src="Views/1/pc/img/common/btn01.png" alt="検索する" onclick="SearchSelectAdr();"/>
```

これを踏まえて、店舗一覧画面へ移動するための処理を先ほどのコードの末尾に追加します。

また移動前の画面（作品ページ）に対する処理をこの画面に対して行わないようにするため、在庫検索画像の走査・クリック処理は一旦コメントアウトしておきます。

```

1 Private Sub getStock()
2     Dim ie As InternetExplorer
3     Set ie = getIE("TSUTAYA") ← TSUTAYAの画面を取得
4     Dim htdoc As HTMLDocument
5     Set htdoc = ie.document ← ドキュメントを取得
6
7     Dim img As HTMLImg ← 画像評価用のオブジェクト変数
8
9     ' For Each img In htdoc.getElementsByTagName("IMG")
10    '     If InStr(img.alt, "店舗を指定して在庫検索") > 0 Then
11    '         img.Click
12    '         Exit For
13    '     End If
14    ' Next
```

① 店頭在庫検索画面を取得します。

② 作品ページに対する処理をコメントアウトします。

```

15
3 16  htdoc.getElementsByName("SearchKey1")(0).Value = "六本木" ← 検索ボックスに店舗名を入力
17
18  Dim searchBtn As IHTMLInputElement
4 19  For Each searchBtn In htdoc.ElementsByTagName("INPUT")
5 20      If InStr(searchBtn.className, "tolCstCondSearchBtn") > 0 Then
21          searchBtn.Click ← 条件に合致したボタンをクリック
22      End If
23  End For
24  Next
25  End Sub

```

- ③ Name属性が「SearchKey1」の要素を取得し、その最初の要素（インデックス番号は0）のvalueプロパティに「六本木」という値を代入します。
- ④ 画面に含まれるすべてのINPUTタグ（フォーム部品）を対象に処理を実行します。
- ⑤ classNameプロパティの値が「tolCstCondSearchBtn」の場合、クリックしてループを抜けます。

これについても、他の属性を評価対象にすることも可能です。処理を実行して、店舗一覧画面に移動できることを確認してください。

5 在庫状況を確認する

店舗一覧画面から店頭在庫状況の表示された作品ページに移動し、在庫状況をメッセージボックスに表示する処理を行います。「六本木」で検索した場合は一覧に複数の店舗が表示されますが、一番上の店舗の在庫状況を表示するようにします。まずは店舗一覧画面でソースを表示して、「在庫を表示する」を検索すると、在庫表示のボタンはhref属性とclass属性を持つaタグであることがわかります。

このうちhref属性は作品・店舗ごとに異なるため、今回はclass属性の値をキーにクリック対象を特定します。値は「zaiko_btn」となっており、いかにも在庫ボタンらしいため適切と思われます。

```

<a href="http://store-tsutaya.tsite.jp/item/rental_dvd/089704291.html?storeId=2000"
class="zaiko_btn">在庫を表示する</a>

```

今回も、先ほど追加した処理をコメントアウトした上で、末尾に処理を追加します。

```
1 Private Sub getStock()  
2     Dim ie As InternetExplorer  
3     Set ie = getIE("TSUTAYA")  
4     Dim htdoc As HTMLDocument  
5     Set htdoc = ie.document  
6  
7     Dim img As HTMLImg  
8  
9     For Each img In htdoc.getElementsByTagName("IMG")  
10        If InStr(img.alt, "店舗を指定して在庫検索") > 0 Then  
11            img.Click  
12            Exit For  
13        End If  
14    Next  
15  
16    htdoc.getElementsByName("SearchKey1")(0).Value = "六本木"  
17  
18    Dim searchBtn As IHTMLElement  
19    For Each searchBtn In htdoc.getElementsByTagName("INPUT")  
20        If InStr(searchBtn.className, "tolCstCondSearchBtn") > 0 Then  
21            searchBtn.Click  
22            Exit For  
23        End If  
24    Next  
25  
26    Dim zaiko_anchor As HTMLAnchorElement  
27    For Each zaiko_anchor In htdoc.getElementsByTagName("A")  
28        If InStr(zaiko_anchor.className, "zaiko_btn") > 0 Then  
29            zaiko_anchor.Click ← 条件に合致したクラス名のリンクをクリック  
30            Exit For  
31        End If  
32    Next  
33 End Sub
```

- ① 店舗一覧画面を取得します。
- ② 画面に含まれるすべてのaタグ(ハイパーリンク)を対象に処理を実行します。
- ③ classNameプロパティが「zaiko_btn」の場合、クリックしてループを抜けます。処理を実行して、在庫状況の表示された作品ページに移動できることを確認してください。

在庫状況は[○]や[x]といった形式で画面下部に表示されています。この情報取得するため、HTMLソースを表示して「在庫状況」で検索してみましょう。在庫状況はclass属性に「state」という値が設定されたdivタグ内に表示されていることがわかります。

```
<div class="state">在庫状況 :<span>[○]</span></div>
```


したがって、画面のspanタグ一式に対してclassNameプロパティを評価するような処理を末尾に追加します。

```
1 Private Sub getStock()  
2     Dim ie As InternetExplorer  
3     Set ie = getIE("TSUTAYA")  
4     Dim htdoc As HTMLDocument  
5     Set htdoc = ie.document  
6  
7     Dim img As HTMLImg  
8  
9     ' For Each img In htdoc.getElementsByTagName("IMG")  
10    '     If InStr(img.alt, "店舗を指定して在庫検索") > 0 Then  
11    '         img.Click  
12    '         Exit For  
13    '     End If  
14    ' Next  
15  
16    ' htdoc.getElementsByName("SearchKey1")(0).Value = "六本木"  
17    '  
18    ' Dim searchBtn As IHTMLElement  
19    ' For Each searchBtn In htdoc.getElementsByTagName("INPUT")  
20    '     If InStr(searchBtn.className, "toICstCondSearchBtn") > 0 Then  
21    '         searchBtn.Click  
22    '         Exit For  
23    '     End If  
24    ' Next  
25  
26    ' Dim zaiko_anchor As HTMLAnchorElement  
27    ' For Each zaiko_anchor In htdoc.getElementsByTagName("A")  
28    '     If InStr(zaiko_anchor.className, "zaiko_btn") > 0 Then  
29    '         zaiko_anchor.Click  
30    '         Exit For  
31    '     End If  
32    ' Next  
33  
34    Dim zaikoLabel As IHTMLElement  
35    For Each zaikoLabel In htdoc.getElementsByTagName("DIV")  
36        If InStr(zaikoLabel.className, "state") > 0 Then  
37            MsgBox zaikoLabel.innerText ← 条件に合致したクラス名のdiv  
38                Exit For ← 要素から文字列を取り出す  
39        End If  
40    Next  
41 End Sub
```

- ① 在庫の表示された作品ページ画面を取得します。
- ② 画面に含まれるすべてのdivタグ（範囲）を対象に処理を実行します。
- ③ classNameプロパティが「state」の場合、メッセージボックスに表示してループを抜けます。

処理を実行して、在庫状況がメッセージボックスに表示されることを確認してください。

6 ワークシートに店舗ごとの在庫を表示する

VBAコード内に直接「六本木」と記述している店舗名をワークシートから取得するようにし、連続処理に対応します。また在庫状況はメッセージボックスではなくワークシートに書き出すようにし、ツールとして完成させます。まずはこれまで作り上げてきたgetStockプロシージャのコメントアウトしてきた部分を元に戻した上で、以下の通り修正してFunctionプロシージャ化します。

```
1 Private Function getStock(ShopName As String, ie As InternetExplorer) As String
2 '   ツタヤの画面を取得
3 '   Dim ie As InternetExplorer
4 '   Set ie = getIE("TSUTAYA")
5
6   Dim htdoc As HTMLDocument
7   Set htdoc = ie.document ← ドキュメントを取得
8
9   Dim img As HTMLImg ← 画像評価用のオブジェクト変数
10
11  For Each img In htdoc.getElementsByTagName("IMG")
12    If InStr(img.alt, "店舗を指定して在庫検索") > 0 Then
13      img.Click ← 条件に合致した画像をクリック
14    Exit For
15  End If
16 Next
17 waitBrowsing ie
18
19 htdoc.getElementsByName("SearchKey1")(0).Value = ShopName ← 検索ボックスに店舗名を入力
20
21 Dim searchBtn As IHTMLElement
22 For Each searchBtn In htdoc.getElementsByTagName("INPUT")
23   If InStr(searchBtn.className, "to1CstCondSearchBtn") > 0 Then
24     searchBtn.Click ← 条件に合致したボタンをクリック
25   Exit For
26 End If
27 Next
28 waitBrowsing ie
```

- ① 店舗名と操作対象画面を引数として受け取り、在庫状況の文字列をリターンするFunctionプロシージャに修正します。あわせて、これまでプロシージャ内でIEを取得する処理を廃止（コメントアウトまたは削除）します。
- ② 各画面移動を行う処理とその後続処理の間に、画面移動完了待ち処理を挿入します。
- ③ 店舗検索テキストボックスに入力する値を、引数として受け取ったShopName変数に変更します。

```

29
30 Dim zaiko_anchor As HTMLAnchorElement
31 For Each zaiko_anchor In htdoc.getElementsByTagName("A")
32     If InStr(zaiko_anchor.className, "zaiko_btn") > 0 Then
33         zaiko_anchor.Click ← 条件に合致したクラス名のリンクをクリック
34     Exit For
35 End If
36 Next
37
2 38 waitBrowsing ie
39
40 Dim zaikoLabel As IHTMLDivElement
41 For Each zaikoLabel In htdoc.getElementsByTagName("DIV")
42     If InStr(zaikoLabel.className, "state") > 0 Then
43         getStock = zaikoLabel.innerText ← クラス名を指定してdivタグから値を取得
44     Exit Function
45 End If
46 Next
47 End Function

```

- ② 各画面移動を行う処理とその後続処理の間に、画面移動完了待ち処理を挿入します。
- ④ 取得した在庫状況を、メッセージボックスで表示するのではなくリターン値として返却するよう変更します。またExit ForからExit Functionに変更します。

この修正を元に、ワークシートにリストアップした店舗名を次々に渡し、返却された在庫状況を転記するメインロジックを作成します。

はじめにワークシートを以下のように作成します。

12			
13	対象店舗	在庫有無	
14	自由が丘		
15	学芸大		
16	祐天寺		
17	中目黒		
18	田町駅前		
19	六本木ヒルズ		
20			
21			
22			
23			
24			
25			

店舗名、在庫有無の列をワークシート上に直接入力して作成する

サンプルでは1列目に店舗名、2列目に在庫有無、14行目からデータが始まるようにしています。この構成にあわせて、モジュール冒頭に以下の通り定数を宣言します。

1	Private Const ROW_START As Long = 14	← 店舗一覧の開始行
2	Private Const COL_SHOPNAME As Long = 1	← 店舗名の列
3	Private Const COL_STOCK As Long = 2	← 在庫状況の列

あとはデータ開始行から店舗名を取得していき、店舗名が取得できなくなるまで繰り返す処理から先ほどの作成したgetStockプロシージャを呼び出すだけです。

1	Public Sub checkAllStock()	
2	Dim sht As Worksheet	
3	Set sht = ActiveSheet	← 店舗一覧のワークシートを取得
4		
5	Dim ie As InternetExplorer	
6	Set ie = getIE("TSUTAYA")	← TSUTAYAの画面を取得
7		
8	Dim i As Long	
9	i = ROW_START	← 行の初期化
10		
11	Do While Trim(sht.Cells(i, COL_SHOPNAME).Value) <> ""	各店舗名をキーに検索処理を実行・結果をセルに転記
12	sht.Cells(i, COL_STOCK).Value = getStock(sht.Cells(i, COL_SHOPNAME).Value, ie)	←
13	i = i + 1	
14	Loop	
15		
16	MsgBox "検索が完了しました"	
17	End Sub	

- ① ワークシートを取得して、変数shtに格納します。これは店舗名の取得や在庫状況の書き出しのために利用します。
- ② TSUTAYAの画面を取得する処理です。元々getStockプロシージャにあった処理と同等です。店舗毎に毎回IEを取得するのは非効率なため、メインロジックで取得したオブジェクトを使いまわすようにします。
- ③ データ開始行から店舗名の列の値をチェックし、空白になるまで処理を繰り返します。
- ④ 店舗名とIE画面を引数にgetStockプロシージャを呼び出し、戻り値をワークシートの在庫有無列に出力します。

以上でツールが完成しました。自分用に近所の店舗名に書き換えるなどして、実際の作品で試してください。最後にP.228を参考に [実行] ボタンを追加しておきましょう。

5-2

TSUTAYA ONLINEで発売予定の作品をチェックする

入力したキーワードでTSUTAYA ONLINEの商品情報を検索し、映像作品の発売日・レンタル開始予定日や書籍の出版年月を一覧化することができるサンプルです。本サンプルを利用することで、お気に入りのタイトルやアーティストの作品の購入・レンタル予定を漏らさず管理することができます。



発売日	レンタル開始日	品名	タイトル
2013年2月	2013年2月	TSUTAYA ONLINE	TSUTAYA ONLINE
2013年2月	2013年2月	TSUTAYA ONLINE	TSUTAYA ONLINE
2013年1月8日	2013年1月8日	TSUTAYA ONLINE	TSUTAYA ONLINE
2013年1月8日	2013年1月8日	TSUTAYA ONLINE	TSUTAYA ONLINE
2013年12月19日	2013年12月19日	TSUTAYA ONLINE	TSUTAYA ONLINE
2013年11月30日	2013年11月30日	TSUTAYA ONLINE	TSUTAYA ONLINE
2012年10月28日	2012年10月28日	TSUTAYA ONLINE	TSUTAYA ONLINE
2012年6月29日	2012年6月29日	TSUTAYA ONLINE	TSUTAYA ONLINE

指定のキーワードの商品を検索して、Excelに発売日・レンタル開始予定日を集める

©角川書店
©2010 第501統合戦闘航空団

Webページの構造を分析して必要な情報を取得しよう

TSUTAYA ONLINEの商品検索結果一覧画面から情報を取得するためには、商品1件のデータの範囲がどこまでか、またその商品のうち必要な情報を取得するためにはどのような方式が最適かなどを検討する必要があります。

この際Webページの構造を十分に分析・理解する必要がありますが、これはTSUTAYA ONLINEに限らず、あらゆるWebサイトについて当てはまることです。このサンプルを参考にWebページの構造に応じて臨機応変に情報の取得方法や入力方法を考案できるようにしましょう。

1 自動化の範囲を決める

まずは手動で発売日・レンタル開始日をチェックする際のの流れを確認し、VBAによる自動化の範囲を決めます。

- ① TSUTAYA ONLINEのトップページ(<http://store-tsutaya.tsite.jp/>)を開きます。
- ② 検索範囲のセレクトボックスから [すべての商品名] または [すべての人名] を選択、検索キーワードを入力して、[検索] ボタンをクリックして検索結果画面に移動します。
- ③ [新しい順] のリンクをクリックして、並び順を変更します。
- ④ ページに表示された各商品について「作品名」「商品ページURL」「発売日」「レンタル開始日」「出版年月」を取得してワークシートに書き出します。
- ⑤ 検索結果が1ページに収まっていない場合、[次へ] のリンクをクリックして検索結果の一覧画面に移動し、④の取得・ワークシート書き出し処理を繰り返します。

この中で特に自動化による恩恵が大きいのは、繰り返し処理を含む④～⑤になります。しかしながら①～③についても技術的難易度・実装量のハードルが低いため、完全自動化によって得られる体感上のメリットを重視して①～⑤まですべての処理を作成することになります。



TSUTAYAのトップページを表示して①、[すべての商品名]を選択して[検索]をクリックする②



検索結果が表示されたら、[新しい順]をクリックする③



商品の一覧が新しい順に並べ替えられる。ここで「作品名」や「発売日」などをExcelに書き出す④



検索結果が複数ページある場合、[次へ]をクリックしてExcelへの書き出し処理を繰り返す⑤

2 ユーティリティ処理の作成

処理の流れを決めたことで、必要となる機能が見えてきました。具体的には、P.159 ①の④以外のすべての処理でWebページ移動を伴うため、Webページ移動の完了待ち機能が必要になります。多くの箇所から参照されることが予想されるため、あらかじめ独立したプロシージャとして実装することにします。

また、コードを書き始める前は、参照設定で「Microsoft Internet Controls」と「Microsoft HTML Object Library」を追加します。参照設定の手順については、「2-2 データ型を明示して入力補完を活用する」の説明を参照してください。

参照 2-2

```
1 Private Sub waitBrowsing(ie As InternetExplorer, Optional FindString As String = "")
2     Dim FoundPos As Long
3
4     Do While FoundPos = 0
5         Do While ie.Busy Or ie.readyState < READYSTATE_COMPLETE
6             DoEvents
7             Loop
8             FoundPos = 0
9             On Error Resume Next
10            FoundPos = InStr(ie.document.body.innerText, FindString)
11            On Error GoTo 0
12            DoEvents
13        Loop
14    End Sub
```

- ① Webページ移動完了待ちプロシージャ（表示文字列検索版）を定義する。
- ② 表示文字列が見つかるまで繰り返します。変数FoundPosが0である場合は処理を繰り返すようにDo～Loop文を作成します
- ③ IEオブジェクトのWebページ移動が完了するまでループします。
- ④ 表示文字列を検索します。引数FindStringが、画面に表示される文字列（Body要素のinnerTextプロパティ）の何文字目に存在しているかInStr関数を使って検索します。

「5-1 TSUTAYA ONLINEでレンタル作品の在庫を店舗ごとに検索する」のサンプルでもWebページの移動完了待ちプロシージャを作成していますが、今回はこれに機能拡張を行います。具体的には、画面に特定文字列が表示されるまで待ち受け処理を再試行するようにしています。なぜこのような機能拡張を行うかについては、P.163の⑤の説明で詳しく触れます。追加した引数はOptionalキーワードをつけており、省略できるようにしておきます（①）。

参照 5-1

②の変数FoundPosには、引数FindStringが画面上に見つかった場合には0よりも大きな値が格納されます。

③ではBusyプロパティとReadyStateプロパティを監視する方法で、移動完了までDo～Loopが繰り返され後続の処理に移動しないようにします。

④のInStr関数は文字列が見つかった場合は、変数FoundPosにその位置を返すため、ループを脱出します。見つからなかった場合は0を返すため、再度③から処理が行われます。なお、空文字列を検索した場合は1を返すため、引数FindStringを省略して呼び出した場合は変数FoundPosには1が格納され、検索処理が再試行されることはありません。

表示文字列の検索処理を「On Error Resume Next」によりエラーを無視するようにしているのは、ワンクッションおいた画面から検索結果一覧画面に自動で書き換わる際にDocumentプロパティにアクセスできずエラーが発生するためです。エラーを無視することで処理全体が停止することなく、また直前でFoundPosに0を格納しているためDoに戻ってから再度待ち受け処理が行われるようになります。

3 TSUTAYA ONLINEのトップページを開く

完全自動化するため、IE画面の新規オブジェクトを作成し、TSUTAYA ONLINEに画面移動するようにします。以下のサンプルコードを実行すると、TSUTAYA ONLINEのトップページが表示されます。

```
1 Public Sub Main()  
2     Dim ie As InternetExplorer  
3     Set ie = CreateObject("InternetExplorer.Application")  
4     ie.Visible = True  
5     ie.navigate "http://store-tsutaya.tsite.jp/"  
② 6     waitBrowsing ie  
7 End Sub
```

IEを起動してNavigateメソッドの引数に指定したURLのWebページへ移動します(①)。Navigateメソッド呼び出し後、P.160の②で作成したwaitBrowsingプロシージャによる移動完了待ち処理を行っています(②)。

4 検索条件を入力して[検索] ボタンをクリックする

検索範囲のセレクトボックス、検索キーワードのテキストボックス、検索実行のボタンの3つが今回の操作対象です。前後の表示文字列などを参考にすると、それぞれ該当するHTMLは次の通りとわかります。


```
<select class="search word select" name="i">
  <option value="201">すべての商品名</option>
  <option value="202">すべての人名</option>
```

```
<input id="searchText" class="search_keyword-input" name="k" placeholder="検索" />
```

```
<input class="search_keyword-btn" value="クエリ送信" type="submit" />
```

これを踏まえて、条件入力・検索実行の処理を以下の通り追加します。この処理を実行するとTSUTAYA ONLINEのトップページが開き、検索条件が入力され検索結果の一覧画面が表示されます。

1	Dim htdoc As HTMLDocument
① 2	Set htdoc = ie.document
3	
② 4	htdoc.getElementsByName("i")(0).Value = "201"
5	
③ 6	htdoc.getElementsByName("k")(0).Value = "ストライクウィッチーズ"
7	
④ 8	htdoc.getElementsByClassName("search_keyword-btn")(0).Click
9	waitBrowsing ie
10	End Sub

- ① HTMLDocumentオブジェクトを取得します。
- ② セレクトボックスの「すべての商品名」を選択します。selectタグのname属性の値がiであることに着目し、これをキーにgetElementsByNameメソッドによりHTML部品要素オブジェクトのコレクションを取得します。
- ③ テキストボックスに検索キーワードを入力します。
- ④ [検索] ボタンをクリックします。

Webページ上の各種部品にアクセスするため、HTML文書のオブジェクトを取得します

(①)。

②では、name属性の値がiの要素は他にないため、ここではインデックス0番目を指定して操作対象のオブジェクトを取得しています。選択肢であるoptionタグがいくつか定義されていますが、ここでは [すべての商品名] を選択するようにします。この選択肢のvalue属性の値は「201」であるため、取得したオブジェクトのvalueプロパティの値に「201」を設定します。なおツールとして完成させる際には選択肢をUIで変更可能にします。

③はセレクトボックス部品の取得と同様に、name属性の値が「k」であることに着目してオブジェクトを取得します。入力する値を取得したオブジェクトのvalueプロパティの値に設定しています。

inputタグのclass属性の値が「search_keyword-btn」であることに着目し、④ではこれをキーにgetElementsByClassNameメソッドを利用してオブジェクトを取得していま

す。alt属性やsrc属性を利用することもできますが、「search_keyword-btn」というクラス名を持つ要素が他に存在しなかったことに加えて、いかにも「検索ボタンらしい値」であることからクラス名による取得としました。検索処理を実行するためにclickメソッドを呼び出して、最後にWebページ移動完了待ち処理を行っています。

5 新しい順に並べ替える

TSUTAYA ONLINEでは、検索結果は「おすすめ順」で表示されるようになっています。今回作成するツールは発売日・レンタル開始日をチェックするツールなので、これを「新しい順」に並べ替えるのが適当でしょう。

そこで、在庫検索結果一覧上部にある並び順のリンクのうち「新しい順」をクリックする処理を追加します。この処理を実行して並び替えが行われるか確認してください。

1	Dim anchorSort As HTMLAnchorElement
① 2	For Each anchorSort In htdoc.GetElementsByTagName("A")
3	If InStr(anchorSort.InnerText, "新しい順") > 0 Then
② 4	anchorSort.Click ← リンクをクリックする
5	waitBrowsing ie
6	Exit For
7	End If
8	Next
9	End Sub

- ① リンクの設定された文字列からクリック対象を検索します。
- ② clickメソッドを利用して条件に合致したリンクをクリックしています。

HTML文書からaタグ一式を取得し、それぞれについてタグに挟まれた文字列（リンクの設定された文字列）であるinnerTextプロパティを評価します（①）。今回は「新しい順」というリンクをクリックしたいので、これを含むかどうかを評価内容にしています。

また後続の処理ですぐにHTML文書を扱えるように、クリック後にWebページ移動完了待ち処理を行っています（②）。クリックしてWebページが移動したらリンクの検索処理を継続する必要がない（継続してはいけない）ため、ループを脱出することにも注意しましょう。

さて、この処理を実行して新しい順に並んだ検索結果一覧画面が表示されたでしょうか？ 実は、この処理はパソコンの性能や通信環境により失敗する場合があります。失敗した場合、リンクがクリックされずに「おすすめ順」のままか、「書き込みができません」などといったエラーメッセージが表示されます。

これはTSUTAYA ONLINEのトップページから検索結果一覧画面を表示する際に、一旦検索結果が表示されていない状態の画面が読み込まれ、その後自動で検索結果一覧を表示する、といったようにワンクッション挟む動作になっているためです。waitBrowsingプロシージャではIEのBusyプロパティがFalseかつReadyStateプロパティが「READYSTATE_COMPLETE」になると移動完了と判断してしまうため、検索結果一覧が「表示されていない画面」の読み込みが完了した時点で後続の処理に進んでしまいます。

このようなWebページの作りに対応するために、waitBrowsingプロシージャに新しく追加した拡張機能を利用します。[検索] ボタンをクリックした後の移動完了待ち処理の呼び出しを以下のように修正しましょう。また、あわせて商品が見つからなかった場合に以降の処理を中止するような処理も追加します。

	1	htdoc.getElementsByClassName("search_keyword-btn")(0).Click
①	2	waitBrowsing ie, "キーワード[" & DataSheet.Cells(ROW_SEARCH_CONDITION, COL_SEARCH_KEYWORD).Value & "]"を含む検索結果"
	3	
	4	If InStr(htdoc.body.innerText, "お客様がお探しの商品は見つかりませんでした") > 0 Then
	5	MsgBox "検索した商品が見つからなかったため処理を終了します"
②	6	Exit Sub
	7	End If
	8	
	9	Dim anchorSort As HTMLAnchorElement
	10	For Each anchorSort In htdoc.getElementsByTagName("A")
	11	If InStr(anchorSort.innerText, "新しい順") > 0 Then
	12	anchorSort.Click
	13	waitBrowsing ie
	14	Exit For
	15	End If
	16	Next
	17	End Sub
	18	----

- ① 「キーワード[<検索キーワード>]を含む検索結果」という文字列の表示を移動完了条件に追加します。
- ② 商品が見つからなかった場合は処理を終了します。

waitBrowsingプロシージャの第2引数に「キーワード[<検索キーワード>]」を含む検索結果」という文字列を設定して呼び出すようにします (❶)。この文字列は検索結果一覧画面のみならず、商品が見つからなかった場合に表示されるため「検索処理の実行が完了し、結果が画面に表示されたこと」を判定するのに適しています。

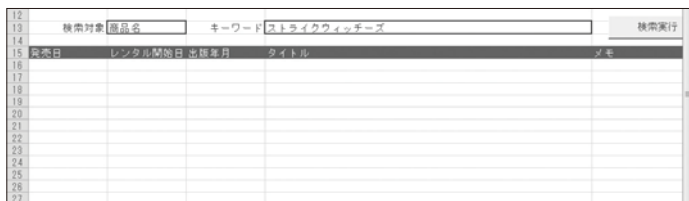
❷では、Body要素のinnerTextプロパティを参照することで画面表示の文字列全体を取得し、ツタヤオンラインのメッセージが含まれるかを確認します。含まれている場合は、メッセージボックスを表示した上で処理を終了します。

これら修正・追加を行った上で処理を実行して、「新しい順」の並べ替えが行われることと商品が見つからない場合は処理が終了することを確認してみましょう。

6 ワークシートを準備する

検索結果一覧の情報を取得・ワークシートに出力する処理を作成していく前に、まずはワークシートのフォーマットを決定しましょう。サンプルでは1つの商品について発売日、レンタル開始日、出版日、商品タイトルの順で1行に書き出すようにしています。またP.161の❷ではVBAの中で直接指定していた検索条件についても、ワークシートで自由に変更できるようにします。具体的には検索対象のセルには入力規則を適用してプルダウンで「商品名」または「人名」が選択できるようにし、検索キーワードはセルに入力するようにしています。Mainプロシージャを呼び出すためのボタンもこのタイミングであわせて配置しておきましょう。

続いて、これらの行や列に関する情報はあとでレイアウトを変更しやすいようにVBAで定数として定義します。またMain以外のプロシージャからもワークシートに対して値の取得・書き出しを行うため、処理対象のワークシートを格納するためのPrivate変数もあわせて宣言します。



12				
13	検索対象	商品名	キーワード	ストライクウィッチーズ
14				検索実行
15	発売日	レンタル開始日	出版年月	タイトル
16				メモ
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				

検索条件を選択・入力する箇所を決定する。またMainプロシージャの呼び出しボタンを配置したり、一覧表が見やすいようにヘッダー行を設けるなどレイアウトを整える

①	1	Private Const COL_RELEASE As Long = 1
	2	Private Const COL_RENTAL As Long = 2
	3	Private Const COL_PUBLISH As Long = 3
	4	Private Const COL_TITLE As Long = 4
	5	
②	6	Private Const COL_SEARCH_TYPE As Long = 2
	7	Private Const COL_SEARCH_KEYWORD As Long = 4
	8	Private Const ROW_SEARCH_CONDITION As Long = 13
	9	
③	10	Private Const ROW_START As Long = 16
	11	
④	12	Private DataSheet As Worksheet

①は各商品の情報を出力するための列定義、②は検索条件入力フォームの列・行定義、③は商品情報を出力する開始行の定義です。④は一覧を作成する処理対象のワークシートを格納するオブジェクト変数です。

続いてこの定義内容に基づいてP.161の④で作成した検索条件の取得処理を変更します。

①	1	Public Sub Main()
	2	Set DataSheet = ActiveSheet : (中略)
②	13	'htdoc.getElementsByName("i")(0).Value = "201"
	14	If DataSheet.Cells(ROW_SEARCH_CONDITION, COL_SEARCH_TYPE).Value = "商品名" Then
	15	htdoc.getElementsByName("i")(0).Value = "201"
	16	Else
	17	htdoc.getElementsByName("i")(0).Value = "202"
	18	End If
	19	
③	20	'htdoc.getElementsByName("k")(0).Value = "ストライクウィッチーズ"
	21	htdoc.getElementsByName("k")(0).Value = DataSheet.Cells(ROW_SEARCH_CONDITION, COL_SEARCH_KEYWORD).Value

- ① ワークシートを取得して、変数DataSheetに格納します。これは検索条件の取得や商品一覧を書き出しのために利用します。
- ② ワークシートから取得した検索対象をセレクトボックスで選択します。
- ③ ワークシートから取得した検索キーワードをテキストボックスに入力します。

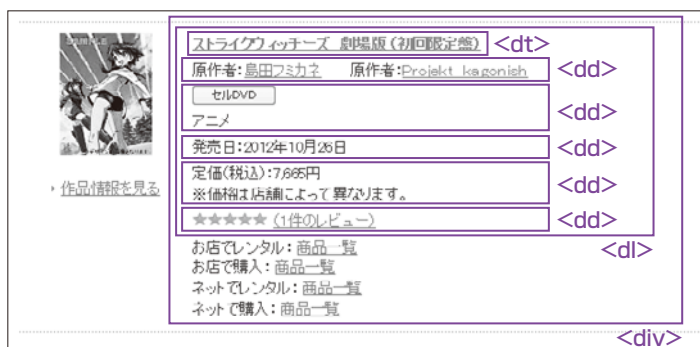
ワークシートの入力内容に応じて、セレクトボックスで選択するvalue属性の値を設定します(②)。サンプルではワークシートで「商品名」が選択されている場合にWebページ上の「すべての商品」を選択し、それ以外の場合は「すべての人名」を選択するようにしています。

③では、ワークシートから取得した検索キーワードを、そのままWebページ上の検索キーワード入力用テキストボックスに入力します。

ワークシートとVBAコードの変更が完了したら、入力内容に応じた検索結果一覧画面が表示されるか確認してみましょう。

7 検索結果の情報を取得してワークシートに書き出す

HTMLを分析すると、各商品の情報はclass属性の値が「detailBox」というdivタグに囲まれていることがわかります。



divタグ(範囲)を親要素と考え、その子要素としてdtタグ(定義リスト)があり、さらにその子要素としてddタグ(定義用語)と複数のddタグ(用語説明)が並んでいる。HTMLソースと画面を見比べてイメージをつかむ

©角川書店

©2010 第501統合戦闘航空団

```
<div class="detailBox">
  <dl>
    <dt>
      <a href="/item/sell_dvd/4582194849895.html">ストライクウィッチーズ
      劇場版(初回限定盤)</a>
    </dt>
    <dd>
      .
      .
      .
      .
      .
      .
  </div>
```

そのため各商品の情報を取得するためには、該当するそれぞれのdivタグに対して処理を行っていくことにします。以下の処理をメイン処理に追加するとともに、各divタグから情報を抽出してワークシートに出力する処理を別途printItemプロシージャとして作成します。

```

1 Public Sub Main()
2     :
3     : (中略)
4     :
5     Dim div As HTMLDivElement
6     For Each div In htdoc.getElementsByClassName("detailBox")
7         printItem div
8     Next
9 End Sub

```

- ① 商品情報の各divタグに対して処理を行います。class属性の値がdetailBoxとなっているdivタグをオブジェクトとして取得し、これを引数にして情報取得・ワークシート出力処理を行うprintItemプロシージャを呼び出します。

次に、Mainプロシージャから呼び出されるprintItemプロシージャは以下の通りです。

```

1 Private Sub printItem(div As HTMLDivElement)
2     Dim dt As HTMLDTElement
3     Set dt = div.getElementsByTagName("DT")(0)
4
5     Dim ItemTitle As String
6     ItemTitle = Trim(dt.innerText)
7     Dim ItemUrl As String
8     ItemUrl = dt.getElementsByTagName("A")(0).href
9
10    Dim row As Long
11    row = ROW_START
12    Do Until DataSheet.Cells(row, COL_TITLE).Value = ""
13        If DataSheet.Cells(row, COL_TITLE).Value = ItemTitle Then
14            Exit Do
15        End If
16        DoEvents
17        row = row + 1
18    Loop
19
20    DataSheet.Cells(row, COL_TITLE).Value = ItemTitle
21    DataSheet.Hyperlinks.Add DataSheet.Cells(row, COL_TITLE), ItemUrl
22 End Sub

```

- ① dtタグをオブジェクトとして取得します。
 ② 商品名を取得します。
 ③ 商品URLを取得します。
 ④ 書き出し対象行を検索します。
 ⑤ 取得した情報をワークシートに書き出します。

①で利用するdtタグとは、「定義する用語」を示すタグです。これに続くddタグで、その用語の説明を記述します。今回はこのdtタグに商品タイトルが格納されているため、まずはこれをオブジェクトとして取得します。今回のケースではdiv内に1つのみ定義されているため、インデックス番号0を指定しています。

ここで取得したdtタグのオブジェクトのinnerTextプロパティの値から商品名を取得して変数ItemTitleに格納します (2)。

3では、商品タイトルに設定されたハイパーリンクを取得するために、dtタグのオブジェクトからaタグを取得します。dtタグ内に1つのみ定義されているため、インデックス番号0を指定しています。取得したaタグのオブジェクトのhrefプロパティの値からURLを取得して変数ItemUrlに格納します。

4では変数rowに書き出し対象行を格納するため、商品タイトル列のセルに値がなくなるまで行番号をインクリメントします。最終的に変数rowには商品情報の最終行の次の行（空白行）の値が格納されます。ただし途中で同名のタイトルが見つかった場合はループを脱出（変数rowにはその商品の行を格納）するようにして、同じ商品の情報を更新できるようにします。

最後に5では、処理対象行の商品タイトル列のセルに商品タイトルを書き込みます。あわせてHyperlinks.Addメソッドによりこのセルに商品ページへのハイパーリンクを設定します。（第1引数はハイパーリンクを設定したいセル、第2引数はハイパーリンク先のURL）

ここまで追加が完了したらMainプロシーダを実行し、タイトルの一覧がハイパーリンクつきで一覧化されることを確認してみましょう。

続いて、タイトル以外に発売日、レンタル開始日、出版年月も書き出せるようにprintItemプロシーダを変更します。

```
1 Private Sub printItem(div As HTMLDivElement)
2     Dim dt As HTMLDTElement
3     Set dt = div.getElementsByTagName("DT")(0)
4
5     Dim ItemTitle As String
6     ItemTitle = Trim(dt.innerText)
7     Dim ItemUrl As String
8     ItemUrl = dt.getElementsByTagName("A")(0).href
9
10    Dim ReleaseDate As String
11    Dim col As Long
```

1 情報格納用変数を宣言します。


```

12 Dim dd As HTMLDDElement
13 For Each dd In div.getElementsByTagName("DD")
14     If InStr(dd.innerText, "発売日:") > 0 Then
15         ReleaseDate = Replace(dd.innerText, "発売日:", "")
16         col = COL_RELEASE
17     ElseIf InStr(dd.innerText, "レンタル開始日:") > 0 Then
18         ReleaseDate = Replace(dd.innerText, "レンタル開始日:", "")
19         col = COL_RENTAL
20     ElseIf InStr(dd.innerText, "出版年月:") > 0 Then
21         ReleaseDate = Replace(dd.innerText, "出版年月:", "")
22         col = COL_PUBLISH
23     End If
24 Next
25
26 Dim row As Long
27 row = ROW_START
28 Do Until DataSheet.Cells(row, COL_TITLE).Value = ""
29     If DataSheet.Cells(row, COL_TITLE).Value = ItemTitle Then
30         Exit Do
31     End If
32     DoEvents
33     row = row + 1
34 Loop
35
36 If DataSheet.Cells(row, COL_TITLE).Value = "" Then
37     DataSheet.Cells(row, COL_TITLE).Value = ItemTitle
38     DataSheet.Hyperlinks.Add DataSheet.Cells(row, COL_TITLE), ItemUrl
39 End If
40 If col > 0 Then DataSheet.Cells(row, col).Value = ReleaseDate
41 End Sub

```

- ② 発売日・レンタル開始日・出版年月を取得します。
- ③ タイトル・URLの上書きを回避します。
- ④ 発売日・レンタル開始日・出版年月をワークシートに書き出します。

①では、発売日・レンタル開始日・出版年月のいずれかを格納する変数 ReleaseDateと、ワークシートへの書き出し対象列を格納する変数 col を宣言します。

HTMLソースや概念図からわかるように、複数の dd タグが定義されています。これらを div タグ内から取得して情報取得対象かの評価を行いますが、今回のケースでは id 属性や name 属性、class 属性などから情報取得対象を特定することができません。

そこで②では、それぞれについて innerText プロパティの値を参照して発売日などの見出しである「発売日:」「レンタル開始日:」「出版年月:」が含まれている場合はこの値を変数 ReleaseDate に格納するようにします。なお、見出しは不要なので Replace 関数を利用して削除しています。

そして、P.163の⑤のように、このツールでは検索結果を新しい順に並べ替えていることから、③では一番最初に書き出されたURLを優先したいため同名タイトルが見つかった場合にも上書きされないようにします。

④では、書き出し対象行・列を指定して変数ReleaseDateの値を書き出します。発売日・レンタル開始日・出版年月のいずれも画面に表示がない場合には変数ReleaseDateに値が格納されないだけでなく変数colの値が0になりエラーとなってしまう(0列目は存在しない)ため、変数colが0より大きい場合にのみ書き出し処理を行うようにします。

処理を実行して、発売日等の情報も含めて一覧化されることを確認してみましょう。

8 次ページ以降の検索結果も取得して書き出す

検索結果が複数ページにわたる場合には次ページ以降の検索結果も取得・書き出しされるように処理を変更します。また検索結果があまりに多い場合に備えてページ数の最大値も設定できるようにします。

次ページが存在するか否かは、「次」という文字列を含むハイパーリンクの有無で判定します。ただし作品名に「次」を含む可能性があるので、あわせてリンク先のURLが検索結果一覧画面のものであるかといったことも判定要素とします。これらの条件を踏まえて、次ページが存在する場合は移動した上で「次ページ有り」という判定結果を返すFunctionプロシーダを以下の通り作成します。

①	1	Private Function NextPageExists(ie As InternetExplorer) As Boolean
	2	Dim anchorNext As HTMLAnchorElement
	3	For Each anchorNext In ie.document.getElementsByTagName("A")
②	4	If InStr(anchorNext.href, "search_result.html") > 0 And anchorNext.innerText = "次"
	5	Then anchorNext.Click
③	6	waitBrowsing ie
	7	NextPageExists = True
	8	Exit Function
	9	End If
	10	Next
	11	End Function

「次」のリンクが見つかった場合はクリックしてWebページを移動し、Trueを返す

- ① 次ページ有無の判定・移動Functionプロシーダを定義する。
- ② HTML文書に含まれる各ハイパーリンクを対象に「次」のリンクを検索する。
- ③ リンクをクリックしてTrueを返す。

判定対象のWebページが表示されたInternet Explorer型の引数ieを受け取り、次ページが存在する場合はTrueを、存在しない場合はFalseを返すNextPageExistsプロシージャを定義します (1)。ハイパーリンク有無の判定のみであればHTMLDocument型のオブジェクトを引数とすることで構いませんが、今回は移動完了待ち処理を含むため上位のInternet Explorer型のオブジェクトを受け取るようにしています。

2では、getElementsByTagNameメソッドによりハイパーリンク式を取得し、それぞれについて評価を行います。hrefプロパティの値に「search_result.html」が含まれるかによりリンク先が検索結果一覧画面であるかを確認し、またinnerTextプロパティの値に「次」が含まれるかにより次ページへ移動するためのハイパーリンクであるかを確認します。

3では、条件に合致するハイパーリンクが見つかった場合にはクリックして、waitBrowsingプロシージャを呼び出すことでWebページの移動完了を待ってから呼び出し元にTrueを返します。ハイパーリンクが見つからなかった場合はFalseを返します。

次にもう1つの繰り返し条件として、P.166の上で解説したモジュール冒頭に取得対象最大ページ数を示す定数MAXPAGENUMの宣言を追加します。サンプルでは値に3を設定しますが、好みに応じて変更しても構いません。

```
1 Private Const MAXPAGENUM As Long = 3
```

最後に、検索結果に次ページが存在し、かつ取得対象最大ページ数に達するまで繰り返し処理を行うようにMainプロシージャを変更します。具体的には各divタグから商品情報を取得・ワークシートに書き出す処理を、NextPageExistsプロシージャの結果とMAXPAGENUMと処理対象ページの大きさを比較した結果と条件にDo~Loop文による繰り返し実行します。

1	Dim pagenum As Long
2	pagenum = 1
3	Do
4	Dim div As HTMLDivElement
5	For Each div In htdoc.getElementsByClassName("detailBox")
6	printItem div
7	Next

1 処理対象ページ番号を初期化します。

2 繰り返し処理を開始します。

3 ページ内の商品情報をワークシートに書き出します。

	8	DoEvents
4	9	pagenum = pagenum + 1
5	10	Loop While NextPageExists(ie) And pagenum < MAXPAGENUM
	11	
6	12	MsgBox "情報の一覧化が完了しました"
	13	End Sub

- 4 ページ番号をインクリメントします。
- 5 次ページの有無とページ番号から繰り返しを判定します。
- 6 処理完了のメッセージを表示します。

①では最初のページとして1を格納します。

②ではDo～Loop文による繰り返し処理の開始を定義します。条件に「次のページが存在するか」を含むため判定は最後に行うこととし、Do～Loop文冒頭での判定は行いません。

③ではP.167の⑦で作成した、各divタグから商品情報を取得してワークシートに書き出す処理です。Do～Loop文の中に入れたため、インデント（行頭位置）を一段右に寄せています。

④では繰り返し条件の判定処理のため、処理対象ページを示す変数pagenumに1を加算します。したがって条件判定時には、変数pagenumは既に関数書き出したページの番号ではなく次に書き出すページの番号を示します。

⑤のNextPageExistsプロシージャの戻り値がTrueであり、また変数pagenum（次ページの番号）が取得対象最大ページ数の定数MAXPAGENUMより小さい場合は処理を繰り返すようにします。

処理が完了したら、その旨メッセージボックスに表示するようにします（⑥）。複数ページにわたって繰り返すような長い処理は完了したのか途中で止まってしまったのかわかりづらいため、必ずメッセージボックス等を使って完了を通知するようにしましょう。

以上でツールは完成です。最大3ページ目まで自動で次のページへ移動しながら、一覧化されることを確認しましょう。最後はP.228を参考に「実行」ボタンを追加しておきましょう。