

[HOME](#) > [即効テクニック](#) > [Excel VBA](#) > [ファイル操作関連のテクニック](#) > サブフォルダを含めてファイル一覧を取得する(Dir関数の再帰呼び出し)

サブフォルダを含めてファイル一覧を取得する(Dir関数の再帰呼び出し) | Excel VBA

ファイル操作関連のテクニック

サブフォルダを含めてファイル一覧を取得する(Dir関数の再帰呼び出し)

(Excel 97/2000/2002/2003/2007)

ツイート

あるプロシージャが、そのプロシージャ内で自分自身を呼び出すような処理を再帰呼び出しと呼びます。

たとえば、簡単にいうなら次のようなパターンです。

```
Sub Sample1()  
    ' 何かの処理  
    :  
    Call Sample1  
    :  
    ' 何かの処理  
End Sub
```

再帰呼び出しを利用するのは「〇〇について××する」というようなケースが多いです。

このうち「××する」部分は共通ですが「〇〇について」の“〇〇”が定まっておらず次々と動的に変化します。

ここでは「“指定したフォルダ”について“存在するファイル名をセルに代入”する」というマクロを、再帰呼び出しで作成してみましょう。

なお、ここでは次のようなフォルダ構造を例にします。

```
C:\Work  
+- Sub1  
    +- tmp  
+- Sub2
```

それぞれのフォルダには、1個以上のファイルが存在するものとします。

さて、任意のフォルダ内に存在するすべてのファイルを調べるには、Dir関数とDo Loopを使って次のようにします。

```

Sub Sample2()
    Dim buf As String, cnt As Long
    Const Path As String = "C:\¥Work"
    buf = Dir(Path & "\¥*.*)"
    Do While buf <> ""
        cnt = cnt + 1
        Cells(cnt, 1) = buf
        buf = Dir()
    Loop
End Sub

```

2行目の「Const Path As String = "C:\¥Work"」で、調査対象のフォルダを指定しています。

いわば「〇〇について」の「〇〇」に該当する部分です。

続く3行目「buf = Dir(Path & "\¥*.*)"」から最後までは、定数Pathで指定されたフォルダ内の全ファイルを抽出する部分です。これが「××する」に該当します。

このままでは、調査対象のフォルダが「C:\¥Work」に固定されてしまいます。

そこで、調査対象のフォルダを、プロシージャの引数として受け取るようにします。

```

Sub Sample3(Path As String)
    Dim buf As String, cnt As Long
    buf = Dir(Path & "\¥*.*)"
    Do While buf <> ""
        cnt = cnt + 1
        Cells(cnt, 1) = buf
        buf = Dir()
    Loop
End Sub

```

これで、たとえば別のプロシージャから

```

Sub Test()
    Call Sample3("C:\¥Work")
End Sub

```

などと呼び出せば、指定したフォルダを調べることが可能です。

次に、Sample3を次のように修正します。

```

Sub Sample3(Path As String)
    Dim buf As String, cnt As Long
    buf = Dir(Path & "\¥*.*)"
    Do While buf <> ""
        cnt = cnt + 1
        Cells(cnt, 1) = buf
    Loop
End Sub

```

```

        buf = Dir()
    Loop
    Pathの中にサブフォルダがあったら
        そのサブフォルダを引数にして自分自身(Sample3)を呼び出す
End Sub

```

あるフォルダの中に存在するすべてのサブフォルダを調べるにはFileSystemObject (FSO) が便利です。
たとえば次のようにします。

```

Sub Sample4()
    Dim f As Object
    With CreateObject("Scripting.FileSystemObject")
        For Each f In .GetFolder("C:¥Work").SubFolders
            Debug.Print f.Path
        Next f
    End With
End Sub

```

実行するとイミディエイトウィンドウに「C:¥Work¥Sub1」と「C:¥Work¥Sub2」が出力されます。
この「Debug.Print f.Path」を「f.Pathを引数にして自分自身(Sample3)を呼び出す」というようにすればいいのですから、Sample3は次のようになります。

```

Dim cnt As Long

Sub Sample3(Path As String)
    Dim buf As String, f As Object
    buf = Dir(Path & "¥*.*)"
    Do While buf <> ""
        cnt = cnt + 1
        Cells(cnt, 1) = buf
        buf = Dir()
    Loop
    With CreateObject("Scripting.FileSystemObject")
        For Each f In .GetFolder(Path).SubFolders
            Call Sample3(f.Path)
        Next f
    End With
End Sub

```

ファイル名をセルに代入するために、変数cntで見つかった個数をカウントしていましたが、その変数cntをグローバル変数に変更してください。

グローバル変数は、前回の値を保持していることがありますので、実行するときは念のために初期化するといいいでしょう。

これで準備完了です。

```

Sub Test()
    cnt = 0

```

```
Call Sample3("C:\Work")  
End Sub
```

で最上位のフォルダ(ここではC:\Work)を指定して実行すれば、存在するサブフォルダを次々と降りていき、存在するファイル名をアクティブシートに書き出します。

フォルダ構造や、実際に行う処理(ここではCells(cnt, 1) = buf)などをあれこれと修正して、動作を確認してください。