

Lesson 1 ▶ プロシージャの呼び出し

使用ファイル

プロシージャ

Point

Sample1プロシージャを実行します。

- ① 3行目の「Call Sample2」で Sample2プロシージャを呼び出しています。
- ② 5行目の「Call Sample3 ("zzz")」で Sample3プロシージャを呼び出しています。

Sample3プロシージャは文字列型の引数を持つプロシージャです。引数とは、プロシージャを呼び出す際に、呼び出し先のプロシージャに渡される値です。呼び出す際に指定した「zzz」が Sample3プロシージャの引数 buf に渡され、Sample3プロシージャ内で使用されています。

標準プロシージャには、Subプロシージャと Functionプロシージャの2種類があります。処理の結果を返す必要がない場合はSubプロシージャを、返す必要がある場合はFunctionプロシージャを利用します。

▶ Subプロシージャの呼び出し

Subプロシージャの呼び出しは、呼び出し元のプロシージャで呼び出し先のプロシージャの処理の結果を利用しないときに使用します。

Subプロシージャを呼び出すにはCallステートメントを使用します。

構文：Call プロシージャ名 または Call プロシージャ名 (引数)

- ◎ 標準モジュールに Sample1、Sample2、Sample3の3つのSubプロシージャを作成します。

```
Sub Sample1 ()
    Debug.Print "aaa"
    Call Sample2-----' ①
    Debug.Print "bbb"
    Call Sample3 ("Sample3へこの文字を引き渡します") -----' ②
End Sub
```

```
Sub Sample2 ()
    Debug.Print "Sample1の2行目から呼び出されました"
End Sub
```

```
Sub Sample3 (buf As String)
    Debug.Print "引数" & buf
End Sub
```

【実行結果】

イミディエイト

```
aaa
sample1の2行目から呼び出されました
bbb
引数Sample3へこの文字を引き渡します
```

Point

▷引数

プロシージャを呼び出す際に、呼び出し先のプロシージャに渡される値です。

名前とデータ型を指定します。複数の引数を指定する場合は「プロシージャ名 (引数1 As 引数1のデータ型, 引数2 As 引数2のデータ型)」のように、「カンマ (,)」で区切って指定します。

戻り値のデータ型を指定します。

▷戻り値

呼び出し先のプロシージャの処理が終わった後、呼び出し元のプロシージャに返される値です。

使用ファイル

プロシージャ

▶ Function プロシージャの呼び出し

Function プロシージャは値を返すことができます。呼び出し元のプロシージャで呼び出し先のプロシージャの処理結果を利用する際に使用します。このとき、呼び出される Function プロシージャを「ユーザー定義関数」と呼ぶこともあります。

構文：

【呼び出し元のプロシージャの記述】

```
変数 = プロシージャ名 (引数) または 変数 = プロシージャ名 ()
```

【呼び出し先の Function プロシージャの記述】

```
Function プロシージャ名 (引数 As 引数のデータ型) As 戻り値のデータ型
    プロシージャ名 = 戻り値
End Function
```

または

```
Function プロシージャ名 () As 戻り値のデータ型
    プロシージャ名 = 戻り値
End Function
```

◎ 次のプロシージャで Function プロシージャの呼び出しの流れを確認しましょう。

```
Sub Sample4 ()                                ' 呼び出し元
    Dim MyInt As Integer

    MyInt = func1 ()                            ' func1 プロシージャを呼び出し
    Debug.Print MyInt
    MyInt = func2 (10)                          ' func2 プロシージャを呼び出し
    Debug.Print MyInt
    MyInt = func3 (5, 10)                       ' func3 プロシージャを呼び出し
    Debug.Print MyInt
End Sub
```

```
Function func1 () As Integer                    ' 呼び出し先
    func1 = 100
End Function
```

```
Function func2 (MyInt As Integer) As Integer   ' 呼び出し先
    func2 = MyInt * 100
End Function
```

```
Function func3 (MyInt1 As Integer, MyInt2 As Integer) As _
    Integer                                     ' 呼び出し先
    func3 = MyInt1 * MyInt2 * 10
End Function
```

【Sample4 実行結果】

イミディエイト

```
100
1000
500
```