

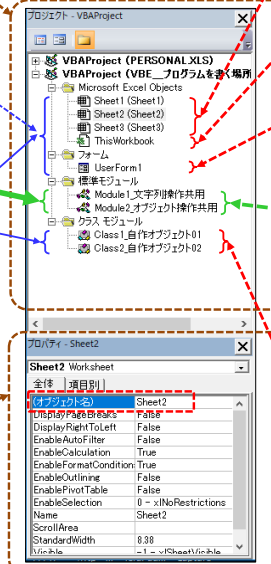
● 「プロシージャ (プログラムの最小単位)」を書く場所についての早見表

★ プロジェクト・エクスプローラ (言わば「モジュールブラウザ」)
「モジュール」と呼ばれる「白紙」たちを見るためのウィンドウです。
フォルダアイコンは各モジュールの「大枠・分類」を表しています。その下の階層の各アイコンは、ひとつひとつが「モジュール」と呼ばれる「白紙状態」すつ単位となっています。
ここで例ですと、「Sheet1」「Sheet2」「Sheet3」「UserForm1」
「ThisWorkbook」「Module1・・・」「Module2・・・」・・・すべてのひとつひとつが「モジュール」です。
「プロシージャ」はプログラムの最小単位ですが、この「モジュール1つの中」に、その複数すつを、書くこととなります。つまり、この「モジュール」が、「プログラムを書く場所」です。また、モジュールごとに役割が異なります。

① この部分だけ、「オブジェクトモジュール」とか、「クラスオブジェクトモジュール」とか呼んだりもするようです。

② 標準モジュール

③ クラスモジュールの仲間



★ プロパティウィンドウ

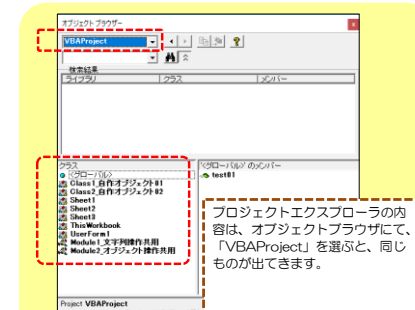
プロジェクト・エクスプローラでクリックしたモノのプロパティ、あるいは、ユーザーフォームのデザイン画面上でクリックした部品のプロパティが表示されます。

ここに表示される「オブジェクト名」は「Nameプロパティ」ではなく「CodeNameプロパティ」で取得することができ、上のペインの「プロジェクトエクスプローラ」では、カッコ付きじゃないほうの名前で表示されます。また、「CodeNameプロパティ」の値は、直接オブジェクト式としても使えます。

例えば「Sheet1」というオブジェクト名 (=CodeNameプロパティの値)のまま変更せしめれば、「Worksheets("Sheet1")」と書くかわりに、「Sheet1」と書くだけでオブジェクト式として成立し、Sheet1のRangeオブジェクトなどを扱えます。なお、「Sheet1」とCodeNameでオブジェクト式を書くとき、(a)シートがどの位置にあっても、また、(b)どんな風Nameプロパティの値を変更しても、Sheet1を確実に明示的に選択 (取得) することが出来ます。

クラスに関するモジュールについては、
・シート1つにつき1つずつ、
・ユーザーフォーム1つにつき1つずつ、
・自作オブジェクト1つにつき1つずつ、
のモジュール (白紙) を、Excelが「自動的に」作ってくれます。プロジェクトエクスプローラの中心。

クラスモジュールやクラスオブジェクトモジュールは、「1つのオブジェクトに紐づいたモジュール」です。もともと、「クラス」は、「オブジェクト」の「設計図」なので。



プロジェクトエクスプローラの内容は、オブジェクトブラウザにて、「VBAProject」を選ぶと、同じものが出てきます。

↓ 各モジュールの種類と、主に何を書く場所か? について

オブジェクトモジュール (クラスオブジェクトモジュール) の中の、「シートモジュール」。
各シートに関連の深いプロシージャや、シート上に直接配置したコマンドボタン等々の「イベントプロシージャ」、その他のイベントプロシージャ」を書く場所です。(基本、各シートの中だけでしか動きません。例えばSheet1のものはSheet2では動きません)

同上の中の、「ブックモジュール」。
ブックに関連の深いプロシージャや、ブックの「イベントプロシージャ」を書く場所です。(基本ブック関係では動きません)

同上(???)の中の、「フォームモジュール」。
フォームに関連したプロシージャや、フォーム上に配置したコマンドボタン等々の「イベントプロシージャ」を書く場所です。(ただし、Excel専用ではなく「Office」専用のクラスオブジェクトっぽい? Wordにもある。よくわかってなくてすみません。基本、フォーム内だけでしか動きません)

「標準モジュール」。
上記のシートモジュール・ブックモジュール・フォームモジュールの各プロシージャから呼び出して使う、あるいは、標準モジュールのプロシージャから呼び出して使う、「共有部品のなプロシージャ」や、グローバルな変数設定、API、などを書くことが多いです。もちろん、どこからも呼び出されずに単独で使うプロシージャも書きます。とにかく一冊プログラム量が多い場所でもあります。

★ リボンの「開発-マクロ」のメニューからマクロを作ると、ここに自動的にその内容 (=コード) が書きこまれます。

「クラスモジュール」。
ここで「自作」の「オブジェクト (単一コレクション)」を作れます。ここは、自作のオブジェクトの「クラス (設計図=定義内容=プログラムそのもの)」を書くための場所です。

この場所をクリックしてからプロパティウィンドウで決めた名前が、自作の「オブジェクトの名前」となります。つまり、その自作のオブジェクトがVBEのコードウィンドウで使えるようになります。このモジュールの中に自作した (内包させた) プロパティやメソッドも、コードウィンドウにて、インテリセンス (自動補完機能) に出てるようになります。

基本、上級者や、企業としてExcelVBA開発をしているような方々がよく使う場所で、我々のような素人はあまり使いたしません。でも使えるほうが間違いに効率が良いそうです。

↓ 各モジュールたちの中に書けるプロシージャの種類

(a) イベントプロシージャ
ただし、シート上に配置したコマンドボタン等々や、シートそのものが保持するイベントに対してのみにしか書けません。
(b) Subプロシージャ (ある意味メインプロシージャ)
(c) 値を返さないFunctionプロシージャ
(d) 値を返すFunctionプロシージャ
※(e)のPropertyプロシージャは書けません。

同上。
ただし、(a)のイベントプロシージャは、ブックが保持するイベントに対してのみにしか書けません。
※(e)のPropertyプロシージャは書けません。

同上。
ただし、(a)のイベントプロシージャは、フォームやフォーム上に配置した部品たち (=コントロールと呼びます) が保持するイベントに対してのみにしか書けません。
※(e)のPropertyプロシージャは書けません。

(b) Subプロシージャ
(★この場所に手作業でーから作ったSubプロシージャは、「マクロ」メニューから直接呼び出せる・ある意味メインプロシージャとなります。上の3つのモジュールのプロシージャやFunctionプロシージャは、「マクロ」メニューからは直接呼び出すことはできません。)

(c) 値を返さないFunctionプロシージャ
(d) 値を返すFunctionプロシージャ
※(a)のイベントプロシージャは書けません。
※(e)のPropertyプロシージャも書けません。

(b) Subプロシージャ (ある意味メインプロシージャ)
(c) 値を返さないFunctionプロシージャ (=値を返さないメソッドの自作)
(d) 値を返すFunctionプロシージャ (=値を返すメソッドの自作)
(e) プロパティプロシージャ (=プロパティの自作)

※(a)のイベントプロシージャは書けません。

注意事項

この3つのモジュールの中に書いたプロシージャは基本、同じモジュールの中のプロシージャやコマンドボタン等々からだけしか呼び出し実行したりできません。(Subも、Functionも、イベントプロシージャも。)
たとえばSheet1モジュールならSheet1モジュールの中のプロシージャやコマンドボタン等々からだけ、Sheet2モジュールならSheet2モジュールの中のプロシージャやコマンドボタン等々からだけ、ThisWorkbookならThisWorkbook、UserForm1ならUserForm1、の中それぞれからだけしか呼び出し実行したりができません。標準モジュールからも呼び出せません。

が、しかし、プロシージャの先頭に「Public」をつけず「Private」を「Public」に書き換えると、標準モジュールや他のモジュールの各プロシージャやコマンドボタン等々から呼び出し実行したりができるようになります。これは、SubプロシージャやFunctionプロシージャだけでなく、イベントプロシージャも同様です。例えば、イベントプロシージャの場合は、Sheet2のプロシージャや標準モジュールのプロシージャから、Sheet1のイベントプロシージャを実行できるようにも・・・ということですが、(この場合『他の場所からSheet1上に作ったコマンドボタンを押したことにする。実際には押してなくても・・・』といったことができます。・・・という意味になります。ユーザーフォームの場合も、フォームの「ShowModal」プロパティを「False」に設定しておくと同じ事が、一応、可能です。ユーザーフォームの目的ボタンを強制的に押したことにできます。実際には押してなくても。)

あと、このクラスオブジェクトモジュールに作られた各モジュールたちは基本、特定のオブジェクトと1対1で紐づいています。また、「Me」キーワードを使って、オブジェクト式を少し短く書くことができます。

以上のごとは、Accessのフォームやレポートでも同じです。Accessのフォームやレポートも「クラスオブジェクト」なので。

注意事項

ここに作られた各モジュールたちは基本、クラスオブジェクトのモジュールたちのように、特定のオブジェクトには紐づいていません。そして、「Me」キーワードは使えません。エラーになります。

Personal.xlsb (または xls) に作った「値を返すFunctionプロシージャ」を、「任意のモジュールのプロシージャから」ではなく、「ワークシートの数式」から呼び出す場合は、Functionプロシージャ名だけ書いたのではセルに「#NAME?」(名前が見つかりませんよ?的な意味?)と表示されてしまっで正常に呼び出せません。その場合は、例えばPersonal.xlsbの場合なら、「Personal.xlsb!Functionプロシージャ名(引数・・・)」と数式に書くことで正常に呼び出せます。

もちろん、他のプロシージャから呼び出す場合なら、そんなことをしなくても通常通り、呼び出せます。

「クラスオブジェクト」の意味

これはAccessの場合も同じです。Accessの場合は、VBEのプロジェクトエクスプローラの中では、「Microsoft Office Access クラスオブジェクト」を見ることがわかります。フォームやレポートがらみのモジュールがあり、それらすべてに1対1で紐づいている (ユーザーが自ら作った) オブジェクトがあります。それが「クラスオブジェクト」です。

クラスオブジェクトの意味としては、「クラス」と呼ばれる設計図をもとに作成されたオブジェクト」というような意味です。ただし、基本、ユーザーが作った「オブジェクト」は全部そうなので、プロジェクトエクスプローラにおいての意味としては、「標準モジュールやクラスモジュールとは違う」という意味とか、「(プログラムの) オブジェクト式の記述に「Me」という語句が使って省略される」とか、「イベントプロシージャが使えるモジュール」といった意味で、とらえればよいと思います。

★ 各プロシージャの簡単な説明

(01) Subプロシージャ (ある意味メインプロシージャ、どのモジュールにも書けます)
一般的なプロシージャです。ExcelやWordの場合は「標準モジュール」に書くことで以下のことできるようになります。(Accessもほとんど同じですが、ほんの少しだけ異なります。その他のOfficeアプリケーションは未チェックです。)
・リボンの「開発」タブの「マクロ」メニュー経由で実行する。
・リボンの自作タブやクイックツールバーのボタンを押下からSubプロシージャの内容を実行する。
・Functionプロシージャと同じように、どこかのプロシージャから呼び出して実行する。

※SubプロシージャではFunctionプロシージャのような「値やオブジェクトを返すプロシージャ」を作ることはできません。

(02) イベントプロシージャ
Subプロシージャの仲間 (派生形) のプロシージャです。形としてはSubプロシージャと同じです。が、実行方法が通常のSubプロシージャとはかなり違うので、「イベントプロシージャ」として明確に分けて説明されることが多いです。(01)の通常のSubプロシージャは後述のような「自動実行」はできません。)
イベントプロシージャは、例えばエンドユーザーが次のような操作をしたときに、「自動実行」されるように設計されています。
・フォームを開いたとき
・シートのセルの値を書き換えたとき
・コマンドボタンを押したとき・・・などなど。
そして更にはそのようなタイミングをきっかけとして、そこからまたさらに、システムが自動的に実行するケースもあるのですが、その場合もイベントプロシージャとして、プログラムを書くことができます。たとえば・・・
・エラーが発生したとき
・セルの数式が自動再計算されたとき・・・などです。
なお、イベントプロシージャを標準モジュールに書くことはできません。

※原則、既定の書き方では、他のプロシージャから呼び出すことができませんが、少し事前準備で書き方を変えることで、他のプロシージャからも呼び出すことができます。これは「イベントが発生してなくても発生したかのごとく強制的に実行できる」という意味です。例えばSheet1に書いたイベントプロシージャを、Sheet2やThisWorkbook、標準モジュールの任意のプロシージャからも呼び出せます。(エラーが出るときは引数が必要な場合が多いので、ダメでもよいので引数を指定しておきます。)

★ 各プロシージャの簡単な説明

(03) Functionプロシージャ
基本的には、Subプロシージャや他のFunctionプロシージャから呼び出して使うプロシージャです。「共有部品」的な役割を持ちます。文字ベースのデータの操作やオブジェクト操作において、頻繁に繰り返すような処理を「共有部品」としてストックしておきます。そのためプロシージャです。
値やオブジェクトを返す (=出力するように作る場合もあれば、返さない (=出力しない) ように作る場合もあります。
「引数」と呼ばれるものを持ち、ワークシート関数と似た形のもので、標準モジュールに書くことが多いですが、もちろん、その他のすべてのモジュールにも書けます。
特に、「値やオブジェクトを返して使うプログラム」を作りたいときは、Functionプロシージャでしか作れないので、そういうプロシージャを作る必要があれば、場所に関係なく作ります。

ただし、Subプロシージャのように「マクロ」メニュー経由で実行したり、クイックツールバーなどのボタンと紐付けて実行することはできません。「リボンや古いタイプのツールバー」のボタンからなら呼び出して実行することができますが、通常メニューからは登録できないためVBAの登録が必要で、未確認です。(もししたらクイックツールバーにもVBAで登録できるのかも知れません)

(04) Propertyプロシージャ
自作の「オブジェクト」を作りたいときに使います。自作の「オブジェクト」に内包させる「プロパティ」を作成する時に使います。ただし、クラスモジュールの中でしか使えません。標準モジュールやクラスオブジェクトモジュールの中では使えません。
===== **※重要な補足** =====
・SubプロシージャではFunctionプロシージャのような「値やオブジェクトを返すプロシージャ」を作ることはできません。
・イベントプロシージャはSubプロシージャの仲間です。
・SubプロシージャもFunctionプロシージャも任意のプロシージャから呼び出すことができます。
・クラスオブジェクト (シートやフォームなど) のモジュールの中にもSubプロシージャやFunctionプロシージャを書くことができます。ただし、その場合は、(既定の書き方では) 書いたモジュールの中しか使えません。例えばSheet1モジュールに書いたSubプロシージャやFunctionプロシージャをSheet2から呼び出すことはできません。