●変数に関する一覧表　※Excel、Word、Access、その他、すべてで共通です。

「生データ（＝素材）」の区分名	「生データ（＝素材）」の特徴	それらが「生データ（＝素材）」を代入・格納するための「変数」の総称	その「変数」を、主にどんな道具を使って、どうやって動かすか	「代入」（＝素材を、変数に格納すること。）の意味するもの	「生データ」の種類と、その種類ごとの「主な特徴」 ※「生データ」の種類＝「生データの”型”」と思ってもらってもOKです。 ※具体的な数値はヘルプなどをご参照ください。	「生データ」を代入するための「変数の種類」の名称	その変数の実際の「型名」	変数宣言の例	「生データ」を変数へ代入する時に必要な記号や語句	「生データ」の別名？	プログラム内でのリテラル値の表現方法 （Excel、Word、Access、その他、すべてで共通）	リテラル値の表現に使う半角の記号を、プログラム内で、「記号」としてではなく「文字列」として表現する方法 （＝Excelに、記号ではなく文字列として認識させる方法）
一般データ	「文字」で書き表せるデータ。 「文字ベース」のデータ。 例えば ・数値 ・文字列 ・日付 ・2値（論理値） など	一般的な総称はありませんが、 当サイトでは、区別しやすくするために、 「一般的な変数」 とか、 「一般変数」 などと呼んでいます。 市販書籍などでも、 そう呼ばれることがあります。	主に、 ・関数（組加工命令） ・演算子（算術記号？四則計算など） ・ステートメント（制御文） といった命令たちを道具として使います。 それらを使って、外部から動かすイメージです。 ※「関数」はオブジェクトを動かす道具としても使うことがありますが、あまり多くはないです。基本的には、「圧倒的に」、文字ベースのデータ（素材）を操作する道具として使います。 ※「ステートメント」は、一般変数も オブジェクト変数も動かせる 「共用の命令」という感じです。 行えることとしましては・・・、 ・変数宣言（変数作成） ・変数への値代入 ・繰り返し（ループ） ・条件分岐（条件の条件設定） ・スリープ（一時停止） ・ジャンプ ・エラー処理 ・・・などがあります。	「代入」 （＝素材を、変数に格納すること。） の意味するもの 「一般変数」の場合の「代入」とは、 例えば、以下のようなことを できるようにするための「準備」です。 ・四則演算（＋、－、×、÷） ・文字の結合 ・文字の切り抜き ・文字の抜き出し（切り出し） ・日付の計算 ・期間算出（出し） ・年齢算出（出し） ・値の合併その他の割り出し 料理に例えると・・・、 野菜や肉などが「素材」 （＝生データ）を、 包丁やナイフ、フライパン、コンロ といった「道具」たち （＝関数、演算子、ステートメント） で料理する（＝加工する）ために、 仮置き用の「小皿・ボウル」（＝変数） に乗せて スタンバイする・・・、 というイメージです。 なので、 「代入」 ＝「素材を仮置き用の小皿に載せる行為」 ＝「料理前の素材のスタンバイ」 ＝「加工前の素材（生データ）のスタンバイ」 というイメージです。 うまい例えができてくすみません。	文字列	文字列型	String	Dim 好きな変数名 As String	代入記号としての「＝」のみ	文字リテラル	半角の「”」または「’」で囲む一例：”abcdefg”　’abcdefg’ 「”」そのものをリテラル用の記号としてではなく、文字そのものとしてプログラム中で認識させたいとき は、シ、書き方が特殊な ので注意が必要です。（右のセルを参照してください。） 「は、Excel やAccessの中のみ中でのみ、又などで条件式を書くときなどに使います が、そのまま”」で囲んで毛柿に何もせずに正常に動作します。 例えば は、SQL文を使は「’」で囲んで、SQL文内の文字列型の値を表現するには「’」で囲みます。 例ーー”SELECT * FROM Tbl WHERE 氏名 = ’鈴木 太郎’;”	”」を文字として認識させたい場合は、「”” 」という感じでダブルクォーテーションを3つつなげて書くか、「” ” ” 」とそのままつなげて書くかで表現できます。 「文字列は”」が記号」を使って結合するときはその必要 なります。変数の値が”」が記号」で結合するときも同様です。逆に、「&記号」を使わない場合は、3つでOKです。） ※詳しくは、http://officetanaka.net/excel/vba/tips/tips90.htm をご参照ください。 ※半角の「’」や全角の「’」や「”」は、通常通り、半角の「’」で囲めばOKです。	なし（もともと何かの記号で値を囲むことがないため）
					数値-超小さい整数 （0～255。この範囲を超えると変数の代入時にエラーとなります。）	バイト型 （あったに使いません。バイナリデータという特別な値を格納するために用意された型だそうです。）	Byte	Dim 好きな変数名 As Byte		数値リテラル？ （整数リテラル 浮動小数点リテラル など）	数字のままでOK一例：123、12345、99.010、など。 ※何かの記号で囲むことはしません。 （ただし、数字を、数値としてではなく、文字列として表現したい場合は文字リテラルの場合と同じようにします。）	
					数値-小さい整数 （約＋3万～－3万。変数代入時のエラーについては同上・同下。）	整数型（Int型） （桁数が少なくてすまエラーになってしまうので、ビジネスシーンでは、特に「集計」では使うことは少ないです。ただ、「＋や－万を超えることはないハズ」、あるかどうか分からないときには、「集計ではなく、ちょっとした計算」には使ってもよ 使用 します。また、ムくてもなく文字型を入力されてうなど者にされたりないなり2値 者にもあって使うことがあります。その辺か、このInt型が使ってあれば他の人がプログラムを見たときに、その機構感がわかるのでそれがいい、という感じかもあるそうです。）	Integer	Dim 好きな変数名 As Integer				
					数値-大きい整数 （約＋21億～－21億）	長整数型（Long型） 「整数を扱うときは、基本、この型を使います。パフォーマンスもInt型よりは良いらのことで、それ良ってで、よく使われます。）	Long	Dim 好きな変数名 As Long				
					数値-超大きい数字 （約＋9万7千～－97万8、少数も少し含む）	通貨型	Currency	Dim 好きな変数名 As Currency				
					数値-超超大きい数字 （千兆以上、少数も少し含む。通常の宣言ができないけど。）	10進型	Decimal	いったんVariant型変数を同じ型で宣言してから、 そのあとCdec関数で指定します。				
					数値-小数を含む値01	単精度浮動小数点型 （ビジネスシーンでは計算エスを多いです。「1掛け」「3掛け」みたいな簡単な計算でも、特に Long型の数値を掛け合わせるとエスることがしばしばいらゆるのセーフーン最終算出などには使用 極点です。） Accessでは絶対にエスるのでパーグン後後では絶対に使ってはいけますんん。Excelのセルの値を代入したときも同じです。 それはれらな便利な使いどこるがあるかもです。科学計算で小数点が第4位までの場合とか？？）	Single	Dim 好きな変数名 As Single				
					数値-小数を含む値02	倍精度浮動小数点型 （ビジネスシーンでは計算エストラブル ほどんど無いです。小数計算は一般的にはこちらの変数が使われます。）	Double	Dim 好きな変数名 As Double				
					日付（時刻も含む）	日付型	Date	Dim 好きな変数名 As Date		日付リテラル	半角の「#」で囲む一例：#2019/01/01#	「”#2019/01/01#” 」といった感じで「#」をダブルクォーテーションで囲めばOKです。 ※全角の「＃」は、通常通り、半角の「’」で囲めばOKです。
					2値（True/False、など、論理値）	ブール型	Boolean	Dim 好きな変数名 As Boolean		ブール-リテラル（？）	基本は「True/False」、場合によっては1/0、や－1/0など。	なし（もともと何かの記号で値を囲むことがないため）
オブジェクト	文字ベースではないデータです。 もっと例えるのが難しいですが、強いて、できるだけ簡単に言うと、 「部品」、あるいは、「ユニット」、「機能の集合体」、という感じです。 「文字ベース（キャラクベース）」との対義語としての 「オブジェクト（モノ）」、 という感じで捉えれることもあります。 もっと言うと「ミニミニロボット」という感じです。 （「操作対象」と言われることもあいますが、それより は、「部品・パーツ、ユニット」「ミニミニロボット」と、とらえて しまヨ方が、イメージ的 機能 になりますし、ヘルプを読むときもラクです。） また、「オブジェクト」には、セル、シート、図形、各種メニュー、などのように、「部品・パーツ・ユニット」として、「画面の中 ” 目で見えるもの ” ともいえ、「画面 で動く目に見えないもの」、もあります。 ただし、「動 画 で動く目に見えないもの」ですから、「機能」にしてもいいし、「部品・パーツ・ユニット」にしても、 「ミニミニロボット」、のほうとなっています。 あと、「オブジェクト」には、 (01)階層構造上下自由 住 支機能（上下階層のオブジェクトの機能を共用できる機能） (02)プロパティ（性格・性質・特性・属性） (03)メソッド（コピー、貼り付け、印刷、といったような、オブジェクトを動かすための命令） (04)イベント（プログラムの自動実行機能） (05)ユーザー入力型（自 動 せ ない、テキストボックスなどに入 力した文字データ等） といった、「5つの機能」が「最後・最考・内包」されています。 特に、(01)と(02)の機能だけは、どのオブジェクトも100%最後・内包 しています。 逆に言うと、(01)と(02)の機能を持つものは100%確実にオブジェクト。 ★超重要注意！！！階層構造とその関連の機能を理解していないと、「中級」には絶対にたどれません。 「オブジェクト」の理解には、「階層構造の理解」が最も重要です。	オブジェクト変数	・階層関連機能 ・プロパティ ・メソッド ・イベント ・列挙（＝Enum、列挙体） ・変数（＝組込変数） ・ステートメント、 （＝関数など） といった命令たちを道具として使います。 基本、 ・階層関連機能 ・プロパティ ・メソッド ・イベント などの命令を、 オブジェクトの例から 引き出して 動かしつつ、 例様に外 例から、 ・列挙（＝Enum、列挙体）、 ・変数（＝組込変数）、 ・ステートメント、 を使って 動かします。 ※「メソッド」は特定のオブジェクトに属 ついた、「そのオブジェクトでしか使えない命令」たちです。	「オブジェクト」の場合の「代入」とは、 「オブジェクト」という「ミニミニロボット」の 「スイッチを入れる」 というイメージの操作です。 料理に例えると、 電子レンジ（いらゆる道具（オブジェクト）を選択して そのスイッチを入れて、 スタンバイする・・・というイメージです。 電子レンジ（いらゆるオブジェクト）の中には あらゆるの、 タイマー機能、ターンテーブル、解凍機能、グリル機能、 スチーム機能、温度調節機能、オーブン機能、 といった、「多 ﾀﾞﾀﾞﾀﾞﾀﾞ（各機能 各）」がスンでいます。 その機能・能力たちを（引き出すかたちで） 思い通りに動かすための道具として、 「スイッチを入れる行為」が、 「オブジェクト」にとっての 「代入」という作業です。 こちらも、うまい例えができなくてすみません。	オブジェクトは （階層構造をかしたもの 各々なんな機能 ）ですけど、初心者のうちは しないほうが無難です。 まず、「オブジェクト」の、このどの「生データ」と呼ぶのは部類 であるか思いますが、そも、一般的なデータ もちろん対象 はしてで 変数操作の観点からは、（特に「オブジェクト変数」の操作の観点から は、 そ 理解して、 まったほうがVBAというものを理解しやすいと思います。 中級にもあがりやすいと思います。 VBAは「中級以上にあがれなかったら、多くがムダになり、最初から やる意味がないリ ですので、そうならないために、このような ご説明をしています。）	オブジェクト型 ※この変数のことを理解できるかどうかが、中級にとがれるかどうかの「大きな分かれ道の一つ」となります。この変数のことを理解すると、オブジェクト操作や各種関数作成、オブジェクトプログラミングヘルプの活用、エラー処理、よりスきな時間・コスト無減などが 使 ﾀﾞﾀﾞﾀﾞﾀﾞ できるようになります。 「For Each」文を使って、数多くのオブジェクトに対する 繰り返し処理（一括処理）にもよく使います。 加えて、一 般データの変数の重要さも理解できるようになります。★★ なので、必ず理解する必要があります。★★ この変数のことを理解するつもりがないなら、VBAはやらないほうが「時間の無駄」にならなくて良いと思います。	Object ※「Range、Worksheet、Chart」、など、より具体的 かつ 明示的な型もあります。 「Object」は、明示的な型がわからないときになんでもOK！ どれでも使える！」的な知ったかぶりをするため、「オールマイティ」な型です。 別 には 量一オブジェクトもコレクションも用方指定 できます。	Dim 好きな変数名 As Object　※オールマイティ Dim 好きな変数名 As Range Dim 好きな変数名 As Worksheet　※単一 Dim 好きな変数名 As Worksheets　※コレクション Dim 好きな変数名 As Chart　※単一 Dim 好きな変数名 As Charts　※コレクション Dim 好きな変数名 As Workbook　※単一	代入記号としての「＝」のほかに、「Set」という語句が必要。 「Set オブジェクト変数 ＝ オブジェクト名」 といった書き方になります。	なし	オブジェクト型変数に代入する際に「リテラル値」はありません。「リテラル値」は一 般的なデータ に対するものです。 が、同様の理 とこれは、「オブジェクト式 そのもの」が、ポジショントの場合のリテラル（＝生データ）に相当するとも思います。 ※ただし、「オブジェクト式」は階層構造を省略しない ほうが無難。ただし、 省略したい 程度的な 事体 ま い の語句（＝部分オブジェクト式）を使ことも あります。	なし（もともと何かの記号で値を囲むことがないため）
一般とオブジェクトの両方					すべての一般データとオブジェクト式	すべての型、オールマイティ型。 一般データ、配列、オブジェクト、、すべてのデータを代入できる変数です。 ただし、値を代入したときに、「基本・自動的に勝手に型の変換を行う」ので、注意が必要です。 例えばExcelだとセルの値が数値の場合、数値系の変数に自動的に型変換されますが、ユーザーフォームのテキストボックスの値を代入した時は、それが数値でも文字列型の変数に勝手に変わってしまいます。これはAccessフォームでも同じです。 そのほか、関数の関り値によってはユーザー的型偏、してなかった 型 になってしまいます。たとえばInputBoxに数値を入力してそれをVariant型の変数に代入してもそ文字列型として扱われます。 そのため、Variant型の変数を扱うときは、ローカルウィンドウやTypeName関数などを使うこと、関数やメソッド、プロパティの関り値やオブジェクト式の型を常に意識・チェックすることが必須です。 Accessでは、Nullを代入できるので、Nullが返ってくる可能性があるときに条件分岐の判定としての変数に使うと、Nullエラー を回避できるので便利です。 **ただ、Excelのセルやvariant型は、小数点処理がにんるるので、VBAコードのそれでの小数点と結果が異なりトラブルが起こることがあるので注意が必要ということです。**	Variant ※オブジェクト型に関しては、「オールマイティ」な のは同じです。	Dim 好きな変数名 As Variant	一般データを扱う場合は代入記号としての「＝」のみ オブジェクトを扱う場合は「Set」が「＝」が必要。	それぞれ	上記のすべて Variant型変数には、上記のすべてのリテラル値みないなオブジェクト式を代入できます。「配列」としても入可能 です の で・・・。	上記に準じます
自作データ					すべての一般データとオブジェクト式	ユーザー定義型	Type	Type	未調査	未調査	未調査	未調査

●重要な関連Webサイト

【VBA入門】徒達しByVal と参照渡しBy Ref の違い・使い方とは
https://www.sejuku.net/blog/71388

VBAはケンだが最強の言語だ
http://shoplaptop.moo.jp/nor-it-salaryman/2017/01/06/vba-strong/

VBA データ型
http://www.keyfound.com/vba/92004

Office TANAKA － 変数の使い方(型は覚えなくていい)
http://officetanaka.net/excel/vba/variable/03.htm

Excel で実数を扱うときの注意　セルの値
http://dekiyaro.sub.jp/ﾌﾟﾛｸﾞﾗﾑ/soft_excel-sample.html

Excel VBA プログラミング　実数を扱うときの落とし穴
http://dekiyaro.sub.jp/ﾌﾟﾛｸﾞﾗﾑ/soft_excel-sample2.html

Excel VBA プログラミング「自動型変換」に関する落とし穴！
http://dekiyaro.sub.jp/ﾌﾟﾛｸﾞﾗﾑ/soft_excel-sample3.html

VBAの型宣言はしたほうがいいかどうか Excel作業をVBAで効率化
https://shakaimen.net/%E9%81%8B%E5%8A%B9%E3%81%A4%E3%81%7%E3%81%8F%E6%8B%9B%E3%81%8C%E3%81%84%E3%81%84%E3%81%8D%E3%81%8B/

※基本、「最低」の「最った」使え方です。　初心者の方は「絶対に」間のみにしないでください。
バカな説明がこの使え方を気例しています。 このwebページでは、InputBoxの部のためだけにチョイスしました。なので InputBoxの部だけを読んでください。

変数の型についてとロータルウィンドウの場合は、最低限のものだけは「絶対に」覚えないと、「自力でのエラー解決」ができません。
「覚えんて覚えなくていい」とは聞きれないように注意ください！

特に、Excelでユーザーフォームを使いたい方や、Access やWordなどとオートメーションで連携をしたい方、また、今 後、他のプログラミング言語も触 したい方は、覚えても、この記事の「覚えんて覚えなくていい」をマネしないでください。
なお、型を先にしないと、たとえばAccessでは計算間違いなどを起こす可 能性もあります。特にフォーム上のテキストボックスの数値をVariant型に代入すると、 勝手にDecimal型になる場合もあれば、 勝手にString型に 当てしまう場合もあります。
Excelでのユーザーフォーム」でも同じようなことが 起こります。テキストボックスの値は、 数値 で あっても「文字列型」に勝手に変換されて、 まいます。
現実には 誤作 用は 起こらないかもしれませんが、そういう思い込み・決めつけをすると、あとで 痛い 目 を見ます。
※ユーザーフォーム上の値で(Variant型をやめて)、0.3をSingle型にして、2900を掛け算すると、「870.0000」という値になってしまいます。正しく計算するには、Single を 型の型にしないといけません。
もちろん あ ま り 使え ま せ ん の は Int型 を か け た 時 だ け に ん り ま せ ん 。
そういったと き に 間違 す る に は 、 や は り 正 し い 「 型 」 の 知識 は 必 要 です 。
「コンピュータは基本、小数点計算などで計算間違いを起すもので、 致命 な ら ど も そ れ が 「 正 常 な 状 態 ” 」 という 病 説 が あ る 以 上 、 「 高 計 」 を 目 的 と す る な ら 、 コ ン ピ ュ ー タ を 信 用 し て は な ら ず 、 正 し い 「 型 」 の 学 習 は 必 須 です 。