

Word VBA, TEXTOS による 言語データ編集

TEXTOS-jp.docx

ver. 2013.11.25

この文書は TEXTOS.xlsm (以下では TEXTOS とします) を簡単に解説したものです。TEXTOS は随時改訂していきますので、この文書も予告なしに改訂していきます。常に最終バージョンを次のサイトにアップロードします。ご使用になられた方はぜひご意見をお寄せください。私のメールアドレスは TEXTOS の開始ページをご覧ください。参考にさせていただき、よりよいものを目指したいと思います。よろしく願いいたします。

<http://lecture.ecc.u-tokyo.ac.jp/~cueda/genko/>

(東京大学・上田博人)

1. 準備

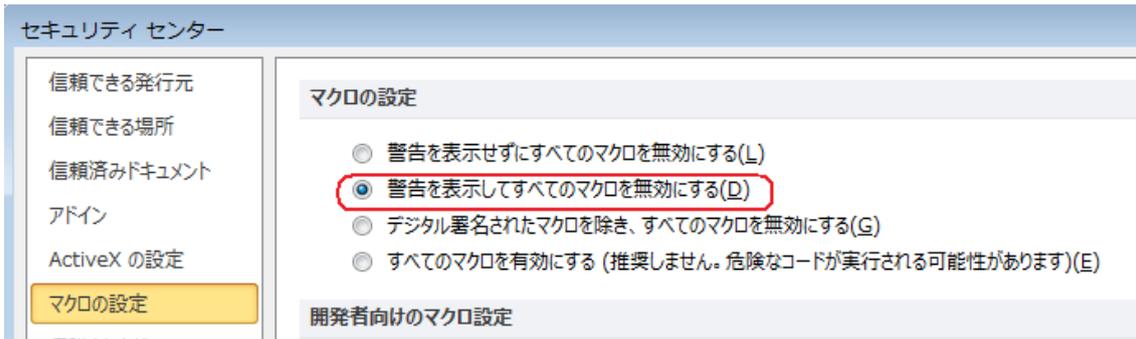


1.1. セキュリティレベル

Word の文書进行处理するためのプログラム言語 VBA (Visual Basic for Application)は「マクロ」というコードを記述することによってコンピュータにさまざまな動作をさせる機能を持ちます。マクロの機能は言語研究に有用ですが、一方、悪意をもった人がこれを悪用すると、コンピュータを勝手に操作してデータを消去したりコンピュータの設定を変えたりすることがあります。このような被害に遭わないように、しかしまた正しい使い方までも禁止しないように、私たち利用者はよく理解して一部の設定を変えなくてはなりません。

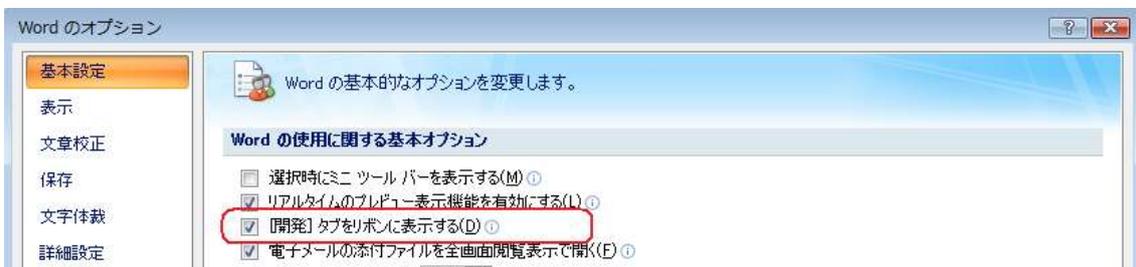
マクロプログラムは Word のセキュリティレベルが最高に設定されていると実行できません。次の手順に従ってセキュリティレベルを1段階下げてください。

* 「ファイル」→「オプション」→「セキュリティセンター」→「セキュリティセンターの設定」→「マクロの設定」（これは Word 2010 の場合です。Word 2007 では「ファイル」→「オプション」を、「Office ボタン」→「Word のオプション」のように読みかえてください。以下でも同様です。）

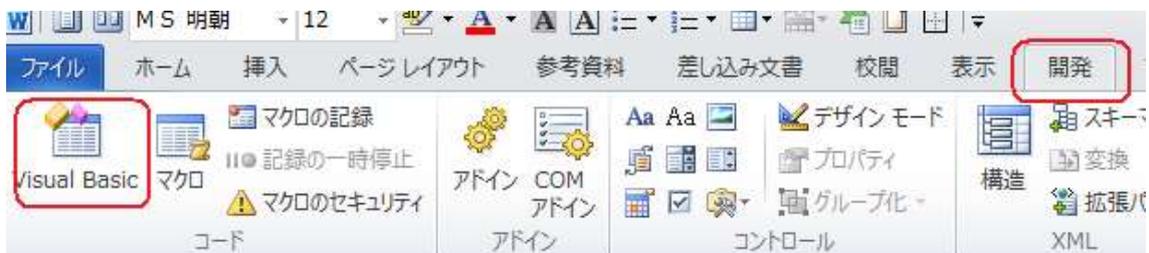


次にすべての Word ドキュメントでマクロプログラムが使用可能になるように設定しましょう。はじめに「開発」タブを表示させます。

*「ファイル」→「オプション」→「基本設定」→「[開発]タブをリボンに設定する」をチェック



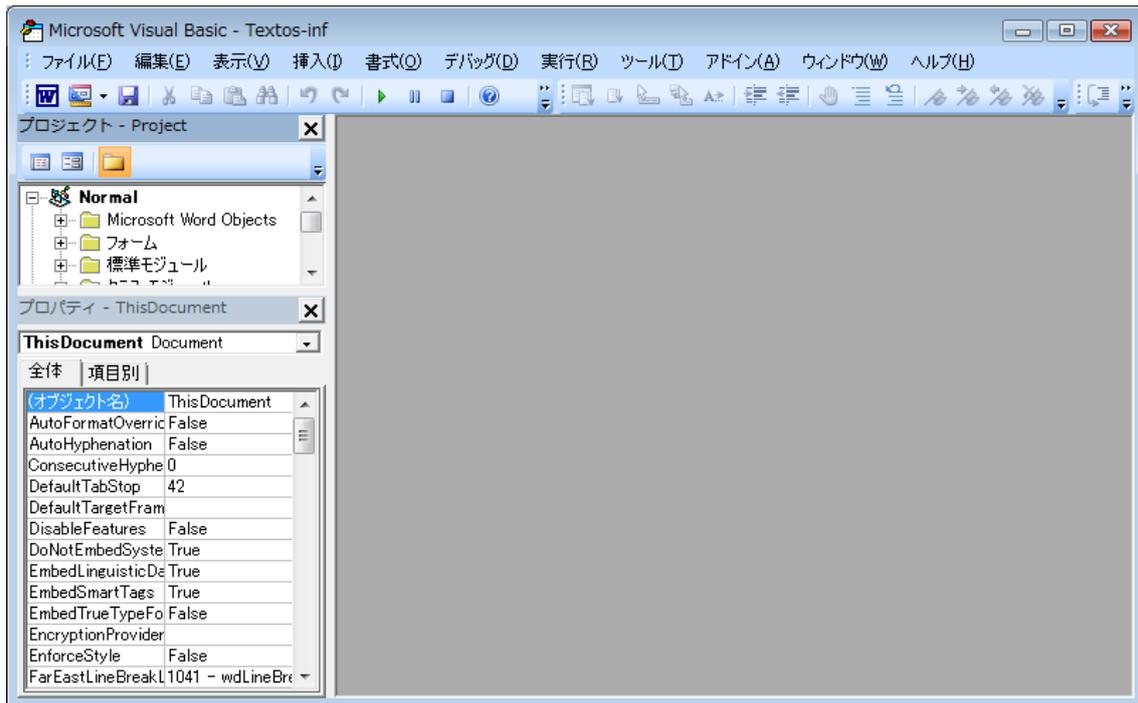
*「開発」タブ→左端の Visual Basic→Visual Basic Editor」が立ち上がります。この操作のショートカットは[Alt]+[F11]です。



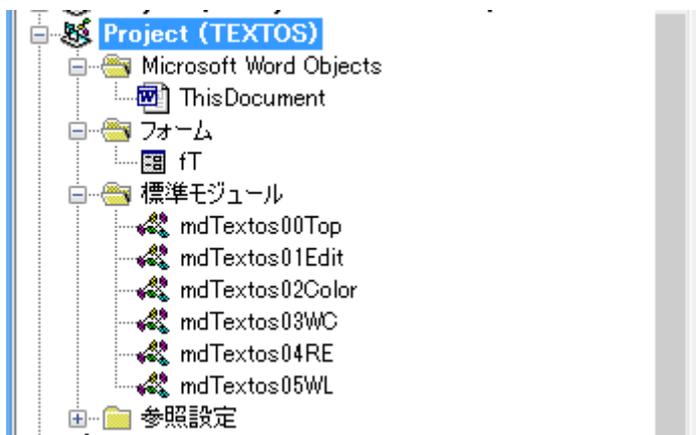
1.2. 汎用化

次の手順に従ってプログラムを Visual Basic Editor の左上の Project Explorer の中にある Normal に登録すれば、すべての Word 文書で使用可能になります。

(1) ショートカット[Alt]+[F11]を同時に押して、Visual Basic Editor を立ち上げてください。

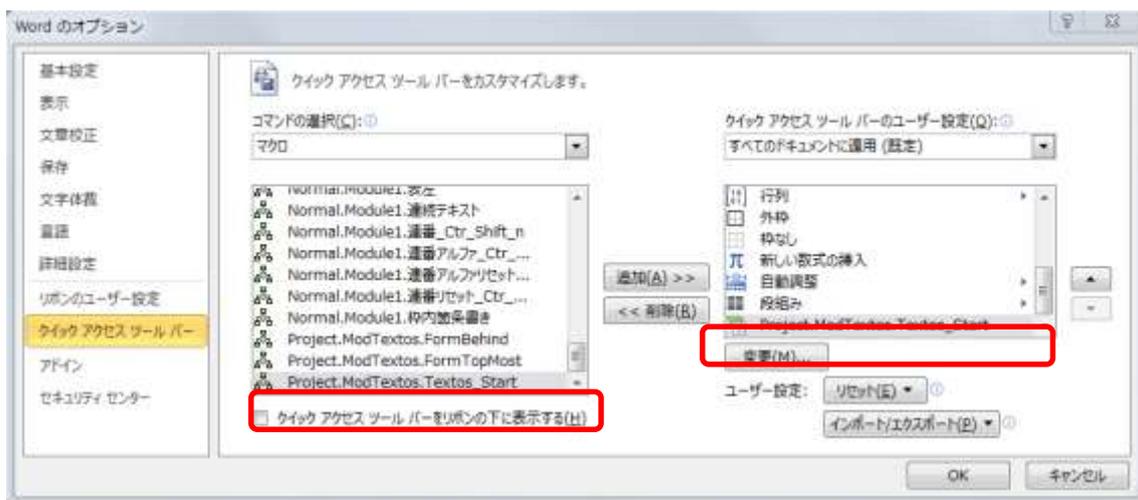


(2) 左上の Project Explorer の Project (Textos)のフォルダー「フォーム」と「標準モジュール」を開き、フォームの fT と標準モジュールのすべての ModTextos を上にある Normal にドラッグ・アンド・ドロップします。



(3) 次に起動プログラム TEXTOS をクイックアクセスツールバーに加えましょう。

* 「ファイル」→「オプション」→「クイックアクセスツールバー」→「コマンドの選択」：マクロ→Normal.TEXTOS を選択→追加（必要ならば「変更」でボタンのアイコンを変えます。）→OK



その結果次のようなボタンが表示されます。



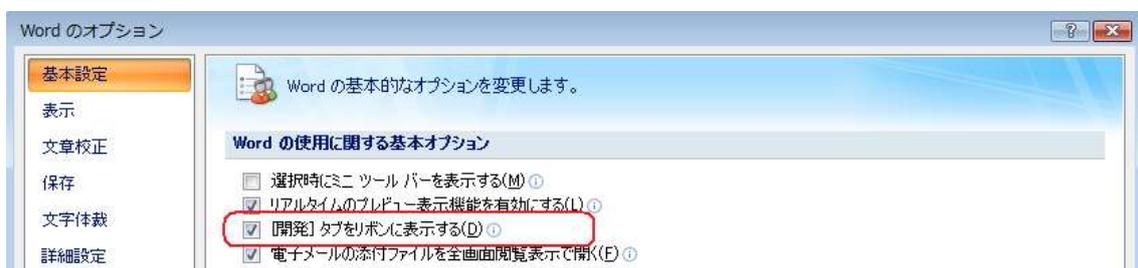
アンインストールするときは、次の手順に従ってください。

*クイックアクセスバーにアイコンを右クリックし、「クイックアクセスバーから削除」をクリック。次に、Visual Basic Editor のプロジェクトペインで該当部分を右クリックし「解放」してください。

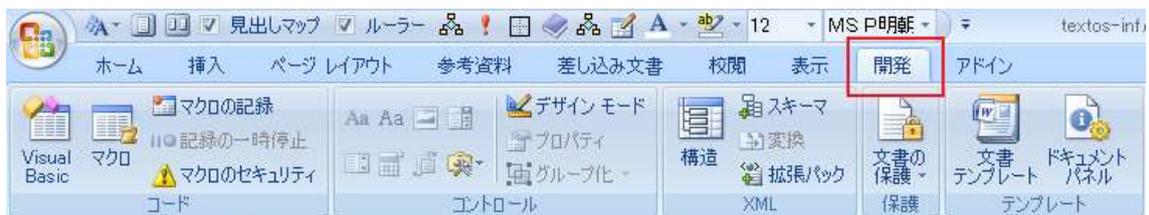
1.2.1. 「開発」タブ

最初に「開発」タブをリボンに設定します。

*「ファイル」→「オプション」→「基本設定」→「[開発]タブをリボンに設定する」をチェック



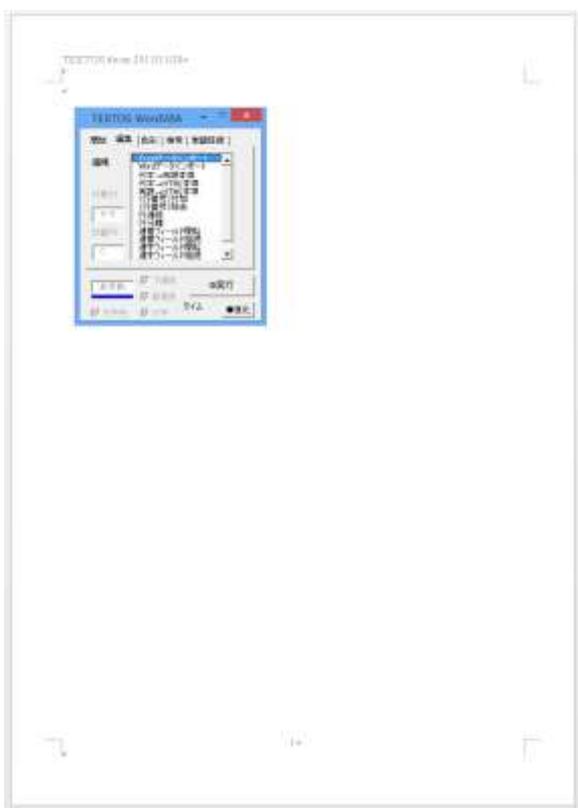
次のようにリボンに「開発」のタブが表示されます。



上のようにリボンの「開発」タブの左端に「Visual Basic」のアイコンが見えます。これをクリックすると「Visual Basic Editor」が立ち上がります。この操作のショートカットは[Alt]+[F11]です。

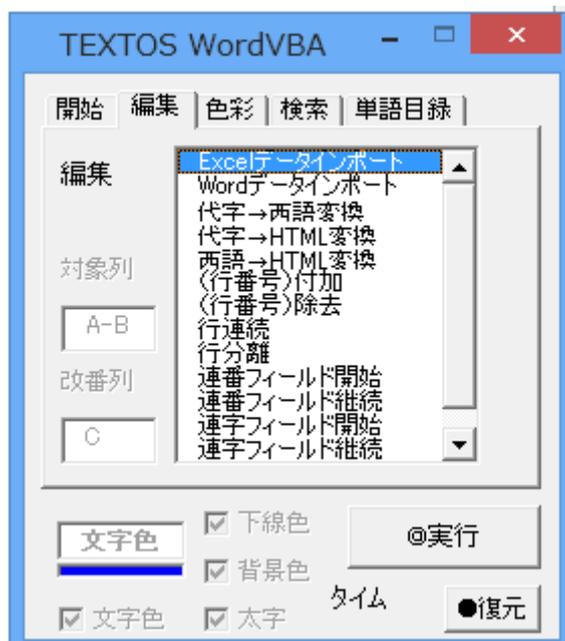
*これで TEXTOS はどの Word ファイルでも使うことができるようになりました。2つのプログラムがあると混乱するので、この TEXTOS.docm は閉じてください。

TEXTOS.docm を立ち上げると次のフッターだけがあるドキュメントが表示されます。このドキュメントを以下の編集・検索・置換の作業に使用します。



2. 編集

[編集]のタブには、さまざまな言語テキスト編集プログラムを集めました。コンボボックスを開いて、選択し、[実行]ボタンを押してください。



2.1. Excel データインポート

次のような Excel シートのデータを Word の文書にインポートします。はじめに、Excel のデータを選択・コピーしてください。コピーのショートカットは Ctrl+c です。

Texto	Tema	Ciudad	Línea
A la recepción de un hotel madrileño llega un profesor extranjero para participar como conferenciante en un seminario sobre Nutrición organizado por una universidad de verano con sede en El Escorial. El profesor hablará con el conserje, pidiéndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la región.	[A] Hotel	(a) Madrid	1
– ¡Buenos días! Desearía una habitación individual para estar tres noches. ¿Qué precio tiene?	[A] Hotel	(a) Madrid	2

次に、Word で TEXTOS を起動し、[編集]の[2.1. Excel データインポート]を実行すると、次の結果になります。

#B: Tema
#C: Ciudad
#D:Línea
Texto

#B: [A] Hotel

#C: (a) Madrid

#D:1

A la recepción de un hotel madrileño llega un profesor extranjero para participar como conferenciante en un seminario sobre Nutrición organizado por una universidad de verano con sede en El Escorial. El profesor hablará con el conserje, pidiéndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la región.

#D:2

- ¡Buenos días! Desearía una habitación individual para estar tres noches. ¿Qué precio tiene?

2.2. 代字→西語変換

次のようなテキストを一定のルールに従ってスペイン語に変換します。(他の言語への変換については5.プログラミングで説明します)

入力

A la recepcio/n de un hotel madrilen/o llega un profesor extranjero para participar como conferenciante en un seminario sobre Nutricio/n organizado por una universidad de verano con sede en El Escorial. El profesor hablara/ con el conserje, pidie/ndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la regio/n.

出力

A la recepción de un hotel madrileño llega un profesor extranjero para participar como conferenciante en un seminario sobre Nutrición organizado por una universidad de verano con sede en El Escorial. El profesor hablará con el conserje, pidiéndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la región.

2.3. 代字→HTML 変換”

入力

A la recepcio/n de un hotel madrilen/o llega un profesor extranjero para participar como conferenciante en un seminario sobre Nutricio/n organizado por una universidad de verano con sede en El Escorial. El profesor hablara/ con el conserje, pidie/ndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la regio/n.

出力

A la recepción de un hotel madrileño llega un profesor extranjero para participar

como conferenciante en un seminario sobre Nutrición organizado por una universidad de verano con sede en El Escorial. El profesor hablará con el conserje, pidiéndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la región.

2.4. 西語→HTML 変換”

入力

A la recepción de un hotel madrileño llega un profesor extranjero para participar como conferenciante en un seminario sobre Nutrición organizado por una universidad de verano con sede en El Escorial. El profesor hablará con el conserje, pidiéndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la región.

出力

A la recepción de un hotel madrileño llega un profesor extranjero para participar como conferenciante en un seminario sobre Nutrición organizado por una universidad de verano con sede en El Escorial. El profesor hablará con el conserje, pidiéndole información sobre los servicios del hotel, así como sobre posibles visitas turísticas por la región.

(以下準備中)

3. 色彩

言語データ分析の結果を表示するとき文字列の背景色や文字色を変えられると便利です。はじめに色彩に使われる変数として、色相(Hue)・明度(Lightness)・彩度(Saturation)と、赤(Red)・緑(Green)・青(Blue)について確認しておきましょう。赤(R)・緑(G)・青(B)は光の三原色です。



3.1. 色彩の性質

色彩について簡単に観察します。はじめに色相について見ましょう。次は、純色（明度(L)=120・彩度(S)=240）で固定し、色相を0から240まで調節して背景色の赤(R)・緑(G)・青(B)の変化を16進数で観察した結果です。RGB値には、赤(R)・緑(G)・青(B)がそれぞれ2桁(00~FF)、全部で6桁の16進数を使います。たとえば青(0000FF)の赤(R)の値は00になり、緑(G)の値は00になり、青(B)の値がFFになります。

色相(H)	明度(L)	彩度(S)	RGB	背景色
0	120	240	0000FF	*****
20	120	240	0080FF	*****
40	120	240	00FFFF	*****
60	120	240	00FF80	*****
80	120	240	00FF00	*****
100	120	240	80FF00	*****

120	120	240	FFFF00	****
140	120	240	FF8000	****
160	120	240	FF0000	****
180	120	240	FF0080	****
200	120	240	FF00FF	****
220	120	240	8000FF	****
240	120	240	0000FF	****

このように色相は0(青)から始まり、240(青)まで続きます。途中のRGB値が0000FF - 00FFFF - 00FF00 - FFFF00 - FF0000 - FF00FF - 0000FFのように変化する円環になります。

次に明度(L)の変化を観察しましょう。はじめに色相=0(青)で実験します。

色相(H)	明度(L)	彩度(S)	RGB	背景色
0	0	240	000000	****
0	20	240	00002B	****
0	40	240	000055	****
0	60	240	000080	****
0	80	240	0000AA	****
0	100	240	0000D5	****
0	120	240	0000FF	****
0	140	240	2B2BFF	****
0	160	240	5555FF	****
0	180	240	8080FF	****
0	200	240	AAAAFF	****
0	220	240	D5D5FF	****
0	240	240	FFFFFF	****

明度(L)=0では赤(R)・緑(G)・青(B)がすべてゼロなので、光の3原色がどれもゼロであるために黒になります。一方、明度(L)=240では光の3原色がどれも最大になるので白になります。純色の青(0000FF)から上方向に黒(000000)に向かうとき、青の数値が減っていきます。一方、純色の青(0000FF)から下方向に白(FFFFFF)に向かうときは赤(R)と緑(G)が同じ値で増えていきます。

次に、青(0000FF)と赤(FF0000)の混色である「紫」(FF00FF)について、明度の変化を観察しましょう。

色相(H)	明度(L)	彩度(S)	RGB	背景色
200	0	240	000000	****
200	20	240	2B002B	****
200	40	240	550055	****
200	60	240	800080	****
200	80	240	AA00AA	****
200	100	240	D500D5	****
200	120	240	FF00FF	****
200	140	240	FF2BFF	****
200	160	240	FF55FF	****
200	180	240	FF80FF	****
200	200	240	FFAAFF	****
200	220	240	FFD5FF	****
200	240	240	FFFFFF	****

紫(FF00FF)から明度(L)を下げて黒(000000)に向かうとき、赤(R)と青(B)の値が同等に減少して、最後は 000000 (黒) となり、逆に、明度(L)を上げて白(FFFFFF)に向かうときは、緑(G)の値が増加して、最後は FFFFFFF (白) になります。

同様に、「青紫」(色相:220)について明度の変化を観察します。

色相(H)	明度(L)	彩度(S)	RGB	背景色
220	0	240	000000	****
220	20	240	15002B	****
220	40	240	430055	****
220	60	240	400080	****
220	80	240	5500AA	****
220	100	240	6A00D5	****
220	120	240	8000FF	****
220	140	240	952BFF	****
220	160	240	AA55FF	****
220	180	240	BF80FF	****
220	200	240	D5AAFF	****
220	220	240	EAD5FF	****
220	240	240	FFFFFF	****

青紫(8000FF)から明度(L)を下げて黒(000000)に向かうとき、赤(R)と青(B)の値がその割合に応じて同等に減少し、最後は 000000 (黒) となり、逆に、明度(L)を上げて白(FFFFFF)

に向かうときは、赤(R)と緑(G)の値がその割合に応じて増加して、最後は FFFFFFF (白) になります。

彩度(S)は、純色を最大(Max)として、同じ明度を保ちながら無彩色に至る度合い (灰色が混じる度合い) を示します。無彩色の赤(R)・緑(G)・青(B)はすべて同じ値です。黒(000000)と白(FFFFFF)は無彩色の中で、それぞれ明度が最小(0)と明度が最大(240)を示します。「青」について彩度の変化を観察しましょう。

色相(H)	明度(L)	彩度(S)	RGB	背景色
0	120	0	7F7F7F	*****
0	120	20	75758A	*****
0	120	40	6A6A95	*****
0	120	60	60609F	*****
0	120	80	5555AA	*****
0	120	100	4A4AB5	*****
0	120	120	4040BF	*****
0	120	140	3535CA	*****
0	120	160	2B2BD5	*****
0	120	180	2020DF	*****
0	120	200	1414EA	*****
0	120	220	0B0BF4	*****
0	120	240	0000FF	*****

このように、彩度を下げていくと (つまり、上表で上昇していくと)、指定された明度が示す無彩色 (灰色) の赤(R)・緑(G)・青(B)値(7F)に向かって、同じ割合でそれぞれの値が減少します。

次のように彩度がゼロであれば、明度に従って赤(R)・緑(G)・青(B)がすべて同値のまま増減します。ここでは青で示しますが、どの色相であっても彩度がゼロであれば同じです。

色相(H)	明度(L)	彩度(S)	赤(R)	緑(G)	青(B)	RGB	背景色
0	0	0	0	0	0	000000	*****
0	20	0	21	21	21	151515	*****
0	40	0	42	42	42	2A2A2A	*****
0	60	0	63	63	63	3F3F3F	*****
0	80	0	85	85	85	555555	*****
0	100	0	106	106	106	6A6A6A	*****
0	120	0	127	127	127	7F7F7F	*****

0	140	0	148	148	148	949494	****
0	160	0	170	170	170	AAAAAA	****
0	180	0	191	191	191	BFBFBF	****
0	200	0	212	212	212	D4D4D4	****
0	220	0	233	233	233	E9E9E9	****
0	240	0	255	255	255	FFFFFF	

次は、青(H=0)、緑(H=80)、赤(H=160)について、それぞれ明度(L)と彩度(S)の変化を示した図です。

青(H=0)

→S ↓ L	0	40	80	120	160	200	240
0	000000	000000	000000	000000	000000	000000	000000
40	2A2A2A	232332	1D1D38	151540	0E0E47	07074E	000055
80	555555	474763	383872	2B2B80	1D1D8D	0E0E9C	0000AA
120	7F7F7F	6A6A95	5555AA	4040BF	2B2BD5	1515EA	0000FF
160	AAAAAA	9C9CB8	8D8DC7	8080D5	7272E2	6363F1	5555FF
200	D4D4D4	CDCDDC	C7C7E2	BFBFEA	B8B8F1	B1B1F8	AAAAFF
240	FFFFFF						

彩度がゼロ(S=0)のときは、明度に従って黒から白のグラデーションができます。彩度を上げると、次第に色が鮮やかになりますが、明度が中位するときにもっとも明確な色になります

緑(H=80)

→S ↓ L	0	40	80	120	160	200	240
0	000000	000000	000000	000000	000000	000000	000000
40	2A2A2A	233223	1D381D	154015	0E470E	074E07	005500
80	555555	476347	387238	2B802B	1D8D1D	0E9C0E	00AA00
120	7F7F7F	6A956A	55AA55	40BF40	2BD52B	15EA15	00FF00
160	AAAAAA	9CB89C	8DC78D	80D580	72E272	63F163	55FF55
200	D4D4D4	CDDCCD	C7E2C7	BFEABF	B8F1B8	B1F8B1	AAFFAA
240	FFFFFF						

赤(H=160)

→S ↓ L	0	40	80	120	160	200	240
0	000000	000000	000000	000000	000000	000000	000000
40	2A2A2A	322323	381D1D	401515	470E0E	4E0707	550000
80	555555	634747	723838	802B2B	8D1D1D	9C0E0E	AA0000
120	7F7F7F	956A6A	AA5555	BF4040	D52B2B	EA1515	FF0000
160	AAAAAA	B89C9C	C78D8D	D58080	E27272	F16363	FF5555
200	D4D4D4	DCCDCD	E2C7C7	EABFBF	F1B8B8	F8B1B1	FFAAAA
240	FFFFFF						

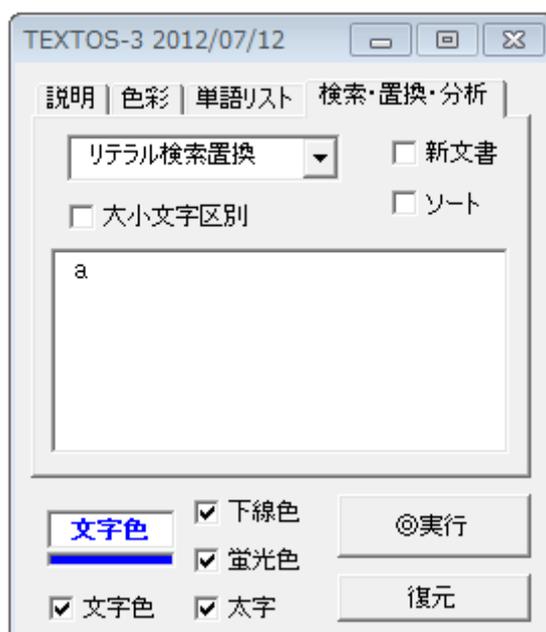
3.2. Word VBA のカラー定数

Word VBA で数字で色の値を指定しますが、その代わりに覚えやすいカラー定数を使うことができます。そのとき、RGB 値は仕様で BGR の順番になるので注意してください。

カラー定数	値	BGR	色見本
wdBlack	1	000000	* * * *
wdBlue	2	FF0000	* * * *
wdTurquoise	3	FFFF00	* * * *
wdBrightGreen	4	00FF00	* * * *
wdPink	5	FF00FF	* * * *
wdRed	6	0000FF	* * * *
wdYellow	7	00FFFF	* * * *
wdWhite	8	FFFFFF	* * * *
wdDarkBlue	9	800000	* * * *
wdTeal	10	808000	* * * *
wdGreen	11	008000	* * * *
wdViolet	12	800080	* * * *
wdDarkRed	13	000080	* * * *
wdDarkYellow	14	008080	* * * *
wdGray50	15	808080	* * * *
wdGray25	16	C0C0C0	* * * *

4. 検索・置換

一定の規則に従って、文書の内容を置換します。置換方式には、リテラル、ワイルドカード、正規表現の3種があります。置換式は「左辺=>右辺」という形式を使います。左辺に検索式、右辺に置換文字を入れてください。



(1) オブジェクト

cbo 検索	「リテラル検索置換」「ワイルドカード検索置換」「単純検索置換」「正規表現検索置換」「検索式頻度」「鍵語頻度」「鍵語外置」「鍵語内置」から選択します。→下の(2)
chk 大小文字区別	大小文字を区別するときはチェックしてください。
chk 新文書 s	新文書に出力するときはチェックしてください。
ソート	Excel に出力した頻度や外置・内置検索の結果を鍵語でソートします。
txt 検索式	検索・置換の検索式を記入します。連立するときは[Ctrl]+Enterで改行してください。最後の式では改行しないでください。(改行すると空白が検索されます。)

(2) コンボボックス(cbo 検索)の選択

単純検索置換	どのような文字列も大小文字を区別してそのまま単純に検索・置換します。
---------------	------------------------------------

正規表現検索置換	Word 文書の書式を除去し、正規表現の規則にしたがって検索・置換します。
リテラル検索置換	Word 文書の書式を保ち、?や*などの記号をそのままリテラルに検索・置換します。大小文字区別を選択できます。
ワイルドカード検索置換	Word 文書の書式を保ち、ワイルドカードの規則にしたがって検索・置換します。
検索式頻度	検索式に一致した文字列の数を出力します。
鍵語頻度	検索式に一致した鍵語の数を出力します。
鍵語外置	検索式に一致した鍵語を段落の外に置いて数を出力します。
鍵語内置	検索式に一致した鍵語を前文脈と後文脈の間に置いて数を出力します。

4.1. ワード検索・置換

Word 文書の中の要素を検索してマークする実験をします。検索の方法は「リテラル」「ワイルドカード」「正規表現」の3種類です。

(a) リテラル検索

「リテラル」では、文書中の文字（列）をそのまま指定して検索文字列強調します。後で見るワイルドカードや正規表現で使う特殊な文字（メタキャラクタ）もエスケープせずにそのまま使えます。たとえば、「?」や「*」などです。検索式をテキストボックスに書き、「実行」ボタンを押します。検索式は後で確認できるように Word などの文書に一度書いておき、それを Copy & Paste するとよいでしょう。とくに英語以外の各国語の文字は直接検索式のテキストボックスに書き込めないなので、そのようにする必要があります。

検索式を連立することもできます。そのときは改行のために[Ctrl]+[Enter]を押してください。または、Word 文書の中で改行したものを貼り付けます。

検索の結果、該当部分が蛍光ペンで示されます。

入力文：

Me voy de aquí dejando parte de mi corazón entre todos mis amigos españoles.

結果：リテラル検索：mi

Me voy de aquí dejando parte de **mi** corazón entre todos **mis** **amigos** españoles.

結果 2 : リテラル検索 : β

βιβλος γενεσεως ιησου χριστου υιου δαβιδ υιου αβρααμ
αβρααμ εγεννησεν τον ισαακ ισαακ δε εγεννησεν τον ιακωβ ιακωβ
δε εγεννησεν τον ιουδαν και τους αδελφους αυτου

「大小文字区別」をチェックするとたとえば「m」は小文字の m だけを検索します。

(b) リテラル置換

1 In thE bEginning God crEatEd thE hEavEns and thE Earth.
2 ThE Earth was without form, and void; and darknEss was on thE facE of thE dEEp.
And thE Spirit of God was hovEring ovEr thE facE of thE watErs.

結果 2: リテラル : e=>E

1 In thE bEginning God crEatEd thE hEavEns and thE Earth.
2 ThE Earth was without form, and void; and darknEss was on thE facE of thE dEEp.
And thE Spirit of God was hovEring ovEr thE facE of thE watErs.

(c) TEXTOS のワイルドカードと正規表現の違い

次は正規表現だけで使用が可能です。

*	直前の文字や式と 0 回以上一致します。
?	直前の文字や式と 0 回または 1 回一致します。
(* * ...)	複数の句をグループ化します。ネストすることができます。

*これらは検索式、置換式の連立式を用いることによって可能になります。一方、ワイルドカードには次の利点があります。

(1) Word の文書の書式を保ったまま検索と置換の処理ができます。

(2) 語の境界 (#, <, >) はラテン拡張文字でも可能です。

たとえば、母音(特殊文字を含む)で始まる語を検索してみましょう。

結果 : ワイルドカード : #¥v%

Es gibt das radikal Böse, aber

ワイルドカードでは特殊文字と一般の母音の間を単語境界と見なさないのが問題なく検索できますが、正規表現では特殊文字と一般の母音の間を単語境界と見なしてしま

うので、後方参照を使います。たとえば、子音で終わる単語を検索するにはワイルドカードを次のように指定します。ここではアポストロフィ(')を含めます。

結果：ワイルドカード：`['a-z]@¥c#`

Cosa c'era **ne'l fior** che m'ai dato?

引用符(') は Word 内でさまざまなコードを持っています。確実に検索するために、Word 内の引用符をコピーして検索・置換のテキストボックスに貼りつけてください。

連続する 2 文字の繰り返しは次のように指定します。

結果：ワイルドカード：`(??)¥1 / 正規表現：(..)¥2`

どんどんと

ワイルドカードでは、`(??)`が 2 文字を示し、これを繰り返すために`¥1` で示します。一方、正規表現では、`(..)`が 2 文字を示し、これを 2 回繰り返すことを示すために`¥2` を使います。

4.2. ワイルドカード検索・置換

(a) ワイルドカード検索

ワイルドカードを使うと一般化した規則によって検索することができます。

任意の文字

?	任意の 1 文字に一致します。たとえば、 <code>b?t</code> で <code>bat, bet, bit, bot, but</code> などを検索します。
*	任意の（無を含めた）文字列に一致します。たとえば、 <code>b*d</code> で、 <code>bd, bad, bed, beed, bread</code> などを検索 します。最左の位置から条件に合った場所で検索を中止します。たとえば、検索式を <code>a*</code> とすると <code>a</code> だけが検索されます。一方、 <code>a*e</code> とすると条件が合うまで（改行位置を超えても）検索を続けます。

単語の先頭と末尾

<	単語の先頭に境界として一致します。たとえば、 <code><in</code> で <code>inactive, interesting, inside</code> などを検索します。全角文字と半角文字の間は単語の境界にはなりません。
>	単語の末尾に境界として一致します。たとえば、 <code>ed></code> で、 <code>called, surprised, visited</code> などを検索します。（全角文字と半角文字の間は単語の境界にはなりません。）

文字セット

[...]	指定した文字のいずれかに一致します。たとえば、b[ai]t で bat と bit を検索します。
[-]	指定した範囲内の任意の 1 文字に一致します。例[n-t]ight で night, right, sight, tight を検索します。範囲は昇順で指定します。×[t-n]
[!x-z]	[...]内の範囲に含まれる文字を除く任意の 1 文字に一致します。例 t[!a-m]ck で tock, tuck に一致しますが tack, tick には一致しません。

繰り返し

{n}	直前の文字または式 n 個に一致します。たとえば、fe{2}d で feed に一致しますが fed には一致しません。
{n,}	直前の文字または式 n 個以上という条件が一致する限り検索します。たとえば、a{1,}で a, aa, aaa, ...に一致します。最長一致をします。
{n, m}	直前の文字または式 n ~ m 個に一致します。たとえば、10{1,3}で 10、100、1000 に一致します。
@	直前の文字または式を 1 個以上という条件に一致します。たとえば、lo@t で lot や loot に一致します。最短一致をします。

*ワイルドカードでは大小文字の区別はできません。A と a を検索するときは[Aa]とします。

*{1,}と@は似ていますが、{1,}は条件が一致する限り最長の文字列を検索し、@は条件が一致する最短の文字列を検索します。そこで、n で始まる単語を検索するときは検索式を<n[a-z]{1,}とします。(検索式を n[a-z]@とすると n の後の 1 文字で一致を終了します。)一方、n で終わる文字列を検索するときは、たとえば検索式を[a-z]@n>のようにします。(検索式を[a-z]{1,}n>とすると、[a-z]{1,}が n を含めた最長一致をしてしまい、さらにその後の n を検索するために一致しません。)

検索式	テキスト
<n[a-z]{1,}	no, not, nation, in, sign
<n[a-z]@	no, not, nation, in, sign
[a-z]{1,}n>	no, not, nation, in, sign
[a-z]@n>	no, not, nation, in, sign

エスケープ文字

特殊文字の検索：(,), [,], {, }, ?, !, *, ¥ を探すときは、その前に¥をつけてエスケープします。たとえば¥?でクエスチョンマークを検索します。(先のリテラル検索ではエス

テープの必要はありません。)

結果 : ワイルドカード : w?s

1	In the beginning God created the heavens and the earth.
2	The earth was without form, and void; and darkness was on the face of the deep. And the Spirit of God was hovering over the face of the waters.

(b) ワイルドカード置換

結果 : ワイルドカード : w?s=>@

1	In the beginning God created the heavens and the earth.
2	The earth @ without form, and void; and darkness @ on the face of the deep. And the Spirit of God @ hovering over the face of the waters.

ワイルドカードによる後方参照は(...)と「¥数字」を使います。数字は検索文字列の(...)の左からの順番に対応します。

結果 : ワイルドカード : (w?s)=>[[¥1]]

1	In the beginning God created the heavens and the earth.
2	The earth [[was]] without form, and void; and darkness [[was]] on the face of the deep. And the Spirit of God [[was]] hovering over the face of the waters.

このように後から検索文字列の一部を参照することを「後方参照」back reference と呼びます。ワイルドカードでは「¥」を使い、正規表現では「\$」を使います。TEXTOS ではワイルドカードでも「\$」を使うことができます。

(c) TEXTOS の拡張ワイルドカード

以下は TEXTOS に固有のワイルドカードです。

¥d	数字[0-9]
¥D	数字以外[!0-9]
¥s	空白文字 (スペース、タブ、改行) [¥t¥n]
¥S	空白文字以外 (スペース、タブ、改行)[! ¥t¥n]
¥w	単語に使用される文字(1文字と一致します。[A-Za-z_0-9])
¥W	単語に使用される文字以外(1文字[!A-Za-z_0-9])と一致します。
¥l	西欧語単語文字[a-zA-Z¥xC0-¥xD6¥xD8-¥xF6¥xF8-¥xFF]

¥L	西欧語単語文字以外[^a-zA-Z¥xC0-¥xD6¥xD8-¥xF6¥xF8-¥xFF]
¥i	キリル文字
¥I	キリル文字以外
¥g	ギリシャ文字
¥G	ギリシャ文字以外
¥e	ハングル
¥E	ハングル以外
¥y	統合漢字
¥Y	統合漢字以外
¥v	母音文字
¥V	母音文字以外
¥c	子音文字
¥C	子音文字以外
¥h	ひらがな[あ-んー]
¥H	ひらがな以外[^あ-んー]
¥k	カタカナ[ア-ンー]
¥K	カタカナ以外[^ア-ンー]
¥z	漢字[¥u4E00-¥uFA6A々々]
¥Z	漢字以外[^¥u4E00-¥uFA6A々々]
¥j	日本語文字[¥u3041-¥uFA6A々々]
¥J	日本語文字以外(1文字[^¥u3041-¥uFA6A々々])

入力文：

次は『旧約聖書』の第1の書『創世記』の第一章第一節「初めに、神は天地を創造された」In the beginning God created the heaven and the earth.のラテン語、中世スペイン語、現代スペイン語の訳である。

- ・ラテン語：In principio creavit Deus caelum et terram.
- ・中世スペイン語：En el principio crió el señor los cielos & la tierra
- ・現代スペイン語：En el principio creó Dios los cielos y la tierra.

結果：ワイルドカード<¥c{1,}> (<:単語境界、¥c:子音)

次は『旧約聖書』の第1の書『創世記』の第一章第一節「初めに、神は天地を創造された」In the beginning God created the heaven and the earth.のラテン語、中世スペイン語、現代スペイン語の訳である。

- ・ラテン語：In principio creavit Deus caelum et terram.
- ・中世スペイン語：En el principio crió el señor los cielos & la tierra

・ 現代スペイン語 : En el principio creó Dios los cielos y la tierra.

4.3. 単純検索・置換

入力文 :

1 In the beginning God created the heavens and the earth.
2 The earth was without form, and void; and darkness was on the face of the deep.
And the Spirit of God was hovering over the face of the waters.

結果 1 : 単純検索 : t (大文字の T を検索しません。)

1 In {**}he beginning God crea{**}ed {**}he heavens and {**}he ear{**}h.
2 The ear{**}h was wi{**}hou{**} form, and void; and darkness was on {**}he face of {**}he deep. And {**}he Spiri{**} of God was hovering over {**}he face of {**}he wa{**}ers.

結果 2 : 単純検索 : t=># (大文字の T を置換しません。)

1 In #he beginning God crea#ed #he heavens and #he ear#h.
2 The ear#h was wi#hou# form, and void; and darkness was on #he face of #he deep.
And #he Spiri# of God was hovering over #he face of #he wa#ers.

4.4. 正規表現検索・置換

(a) 正規表現検索

正規表現は複雑な文字列処理に適していますが、その出力は Word の文字装飾は消失して、プレーンテキストになります。書式を保ちながら置換するときは次のワイルドカードを使用してください。

特殊文字

¥t	水平タブに一致します。
¥b	任意の単語境界に一致します。
¥B	任意の単語境界以外の位置に一致します。
¥n	改行に一致します。

入力文 : 最後の改行コードを含めないように選択してください。

The Universal Declaration of Human Rights Article 1. All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act

towards one another in a spirit of brotherhood.

結果 : 正規表現: `¥bin¥b`: 単語境界に挟まれた in

The Universal Declaration of Human Rights Article 1. All human beings are born free and equal `{*in*}` dignity and rights. They are endowed with reason and conscience and should act towards one another `{*in*}` a spirit of brotherhood.

beings の中の in は、単語境界に挟まれていないので一致しません。

文字クラス

<code>[xyz]</code>	文字セットに含まれている任意の 1 文字に一致します。[...]の中では., ?, *などをエスケープする必要はありません。
<code>[^xyz]</code>	文字セットに含まれていない任意の 1 文字に一致します。
<code>.</code> (ピリオド)	改行(<code>¥n</code>)以外の任意の文字に一致します。
<code>¥w</code>	単語に使用される任意の文字(アルファベット、数字、アンダースコア) <code>[a-zA-Z0-9_]</code> に一致します。
<code>¥W</code>	単語に使用される文字以外の任意の文字に一致します。 <code>[^a-zA-Z0-9[a-zA-Z_0-9]</code> と同じです。
<code>¥d</code>	任意の数字に一致します。 <code>[0-9]</code> と同じです。
<code>¥D</code>	任意の数字以外の文字に一致します。 <code>[^0-9]</code> と同じです。
<code>¥s</code>	任意のスペース文字に一致します。 <code>[¥t¥r¥n¥f]</code> と同じです。
<code>¥S</code>	任意の非スペース文字に一致します。 <code>[^ ¥t¥r¥n¥f]</code> と同じです。

入力文 :

All human beings are born free and equal in dignity and rights.

結果 正規表現検索: `[e-h]` (e, f, g, h, i に一致)

All `{*h*}`uman b`{*e*}`in`{*g*}`s ar`{*e*}` born `{*f*}`r`{*e*}``{*e*}` and `{*e*}`qual in di`{*g*}`nity and ri`{*g*}``{*h*}`ts.

結果 正規表現検索: `[^a-v]` (a-v 以外に一致)

All`{* *}`human`{* *}`beings`{* *}`are`{* *}`born`{* *}`free`{* *}`and`{* *}`equal`{* *}`in`{* *}``{*}`dignit`{*y*}``{* *}`and`{* *}`rights`{*.*}`

結果 正規表現検索:[c-i] (c, d, e, f, g, h, i に一致)

All human beings are born free and equal in dignity and rights.

結果 正規表現検索:[^c-i] (c, d, e, f, g, h, i 以外に一致)

All human beings are born free and equal in dignity and rights.

選択、グループ化、繰り返し

	複数の句を1つの正規表現にまとめ、これらのうちの任意の句に一致します。たとえば、d(os a)は dos または da に一致します。
(...)	複数の句をグループ化して1つの句を作成する。(ab)*c は abc または c に一致します。
+	1個以上の直前の文字に一致します。{1,}と同じです。e+で e, ee, eee, ... に一致します。
*	ゼロ個以上の直前の文字に一致します。{0,}と同じです。ah*で a, ah, ahh, ... に一致します。
?	ゼロ個または1個の直前の文字に一致します。{0,1}と同じです。books?で book と books に一致します。
{a}	先行する正規表現 a 個に一致します。[aeiou]{2}で2母音の連続(ei, ee, ua など)に一致します。
{a,}	先行する正規表現 a 個以上の直前の文字に一致します。[aeiou]{3,}で3母音の連続(aei, uai, auuu など)に一致します。
{a,b}	先行する正規表現 a 個以上、b 個以下に一致します。[aeiou]{2,4}で2-4母音の連続(ei, aei, uai, auuu など)に一致します。

結果 正規表現検索 (free|equal) (free と equal に一致)

All human beings are born {*free*} and {*equal*} in dignity and rights.

結果 正規表現検索 [e-h]+ ([e-h]の連続に一致)

All {*h*}uman b{*e*}in{*g*}s ar{*e*} born {*f*}r{*ee*} and {*e*}qual in di{*g*}nity and ri{*gh*}ts.

結果 正規表現検索 [aeiou]{2} (2母音の連続)

All human b{*ei*}ngs are born fr{*ee*} and eq{*ua*}l in dignity and rights.

結果 3 正規表現検索 `[^aeiou]{2,}` (母音以外の文字 2 個以上の連続に一致)

```
A{*ll h*}uma{*n b*}ei{*ngs *}are{* b*}o{*rn fr*}ee a{*nd *}equa{*l *}i{*n
d*}i{*gn*}i{*ty *}a{*nd r*}i{*ghts.*}
```

(b) 正規表現置換

正規表現の後方参照を使うと、検索式の一部を参照することができます。句を括弧で囲み、\$の後に 1 つの数字を続けることによってその句を指定します。

置換トークン

\$n	検索パタンの n 番目の(...)に一致した文字列
\$&	検索パターンに一致した文字列全体
\$`	検索パターンに一致した文字列の前にある文字列
\$'	検索パターンに一致した文字列の後にある文字列
\$_	対象文字列全体
\$\$	\$という文字

入力文 :

```
Rumi: Hola, profesor.Prof. Rubio: Buenos di/as.Rumi: Buenos di/as. Nos encontramos ahora
en la Universidad [[Complutense]] de Madrid. ??Do/nde nos vamos ahora?
Prof.: Bueno, vamos a iniciar hoy el [[Camino]] del [[Cid]], la primera parte.
```

結果 : 正規表現 : HTML コードを作成します。

```
a/=>&aacute;
e/=>&eacute;
i/=>&iacute;
o/=>&oacute;
u/=>&uacute;
```

結果 : 正規表現 : `([aeiou])/=>&$1acute;`; 上の連立式を折りたたみます。

```
Rumi: Hola, profesor.Prof. Rubio: Buenos d&iacute;as.Rumi: Buenos d&iacute;as. Nos
encontramos ahora en la Universidad Complutense de Madrid. ??D&oacute;nde nos vamos
ahora? Prof.: Bueno, vamos a iniciar hoy el Camino del Cid, la primera parte.
```

結果：正規表現：#(c%)=>[[\$1]] (c で始まる語を[...]でマークします。)

Rumi: Hola, profesor.Prof. Rubio: Buenos días.Rumi: Buenos días. Nos encontramos ahora en la Universidad [[Complutense]] de Madrid. ¿Dónde nos vamos ahora?
Prof.: Bueno, vamos a iniciar hoy el [[Camino]] del [[Cid]], la primera parte.

次は中世スペイン語の資料を文字化した資料です。

Otro(22)ssí mando que los menestrales non echen suerte en el judgado por seer juezes, ca el juez deve tener la seña, e tengo que si <a> afruenta viniessse o a logar de periglo e omne vil o rafez toviessse la seña que podrié (23) caer el concejo en grant onta e en grant vergüença.

(22)は語の途中で改行され ssí 以下が 22 行目になることを示しています。文法研究のためには、これを Otróssí (22)にする必要があります。これは次の置換式によって実現できます。

結果：正規表現：(¥(¥d+¥))(&)=>\$2 \$1

Otróssí (22) mando que los menestrales non echen suerte en el judgado por seer juezes, ca el juez deve tener la seña, e tengo que si <a> afruenta viniessse o a logar de periglo e omne vil o rafez toviessse la seña que podrié (23) caer el concejo en grant onta e en grant vergüença.

(c) エスケープ文字

特殊文字の検索 (,), [,], {, }, ?, !, .(ピリオド), +, *, |, ¥を探すときは、その前に¥をつけてエスケープします。たとえば¥?でクエスチョンマークを検索します。

入力文：

¿Cómo está usted?

結果：正規表現 ¥? (クエスチョンマーク)

¿Cómo está usted?

(d) 後方参照

後方参照を使うことで式の内容を記憶させ、それを後から参照させることができます。

(...)¥n	検索文字列の(...)の式に一致した文字列が記憶され、それを n 回繰り返して参照します。
(...)...(...)=>\$n	検索文字列の(...)の式に一致した文字列が記憶され、置換文字列でそれを参照して再生します。n は(...)の順番に対応する番号で

	す。
--	----

入力文 :

どんどんテーマが広がって、ますます興味がわいてきた。

結果： 正規表現：(..)¥2:2つの同じ文字の連続

どんどんテーマが広がって、ますます興味がわいてきた。

参照する文字（列）がわかっているときは、検索式をたとえば「どん{2}」のようにすることができますが、ここではそれが他にも「ますます」、「ぐんぐん」のように、さまざまに変化する場合を想定しています。

(e) TEXTOS の拡張正規表現

TEXTOS では正規表現を拡張して次の特殊文字を再定義します。

#	単語の境界：#b%は b で始まる単語を検索します。
&	単語文字 1 個以上 ¥I+
%	単語文字 0 個以上 ¥I*
¥I	西欧語単語文字 [a-zA-Z¥xC0-¥xD6¥xD8-¥xF6¥xF8-¥xFF]
¥L	西欧語単語文字以外 [^a-zA-Z¥xC0-¥xD6¥xD8-¥xF6¥xF8-¥xFF]
¥i	キリル文字
¥I	キリル文字以外
¥g	ギリシャ文字
¥G	ギリシャ文字以外
¥e	ハングル
¥E	ハングル以外
¥y	統合漢字
¥Y	統合漢字以外
¥v	母音文字
¥V	母音文字以外
¥c	子音文字
¥C	子音文字以外
¥h	ひらがな [あ-んー]
¥H	ひらがな以外 [^あ-んー]
¥k	カタカナ [ア-ンー]
¥K	カタカナ以外 [^ア-ンー]
¥z	漢字 [¥u4E00-¥uFA6A々々]

¥Z	漢字以外 [^¥u4E00-¥uFA6A々々]
¥j	日本語文字 [¥u3041-¥uFA6A々々]
¥J	日本語文字以外 [^¥u3041-¥uFA6A々々]

単語の境界

次は連立検索置換式を使って、はじめに上の正規表現で(..)の数字を単語の外に置き、その後、oで始まる単語(#o%)を外置で検索した結果です。

Otro(22)ssí mando que los menestrales non echen suerte en el juzgado por seer juezes, ca el juez deve tener la seña, e tengo que si <a> afruenta viniessse o a logar de periglo e omne vil o rafez toviessse la seña que podrié (23) caer el concejo en grant onta e en grant vergüença.

結果：正規表現：

(¥(¥d+¥))(&)=>\$2 \$1

#o%

{*Otrossí*} (22) mando que los menestrales non echen suerte en el juzgado por seer juezes, ca el juez deve tener la seña, e tengo que si <a> afruenta viniessse {*o*} a logar de periglo e {*omne*} vil {*o*} rafez toviessse la seña que podrié (23) caer el concejo en grant {*onta*} e en grant vergüença.

結果：正規表現：#(m%)=>[\$2]（語頭の単語境界#を使った置換式の置換文字列のトークンの数字は+1としてください。）

Otro(22)ssí mando que los [menestrales] non echen suerte en el juzgado por seer juezes, ca el juez deve tener la seña, e tengo que si <a> afruenta viniessse o a logar de periglo e omne vil o rafez toviessse la seña que podrié (23) caer el concejo en grant onta e en grant vergüença.

結果：正規表現：(%[rs])#=>[\$1]（語尾の単語境界#を使った置換式の置換文字列のトークンの数字はそのまましてください。）

Otro(22)ssí mando que [los] [menestrales] non echen suerte en el juzgado [por] [seer] [juezes], ca el juez deve [tener] la seña, e tengo que si <a> afruenta viniessse o a [logar] de periglo e omne vil o rafez toviessse la seña que podrié (23) [caer] el concejo en grant onta e en grant vergüença.

日本語文字

日本語文字を検索するときは、¥h（ひらがな）、¥k（カタカナ）、¥z（漢字）、¥j（日本語文字）を使用します。

入力文：

親譲りの無鉄砲で小供の時から損ばかりしている。

結果：正規表現 ¥z{2} (漢字 2 文字の連続)

{*親譲*}りの{*無鉄*}砲で{*小供*}の時から損ばかりしている。

結果：正規表現:¥z+ (1 個以上の漢字)

{*親譲*}りの{*無鉄砲*}で{*小供*}の{*時*}から{*損*}ばかりしている。

入力文：

『坊っちゃん』夏目漱石

親譲りの無鉄砲で小供の時から損ばかりしている。小学校に居る時分学校の二階から飛び降りて一週間ほど腰を抜かした事がある。なぜそんな無闇をしたと聞く人があるかも知れぬ。別段深い理由でもない。新築の二階から首を出していたら、同級生の一人が冗談に、いくら威張っても、そこから飛び降りる事は出来まい。弱虫や一い。と囃したからである。小使に負ぶさって帰って来た時、おやじが大きな眼をして二階ぐらいから飛び降りて腰を抜かす奴があるかと云ったから、この次は抜かさずに飛んで見せますと答えた。

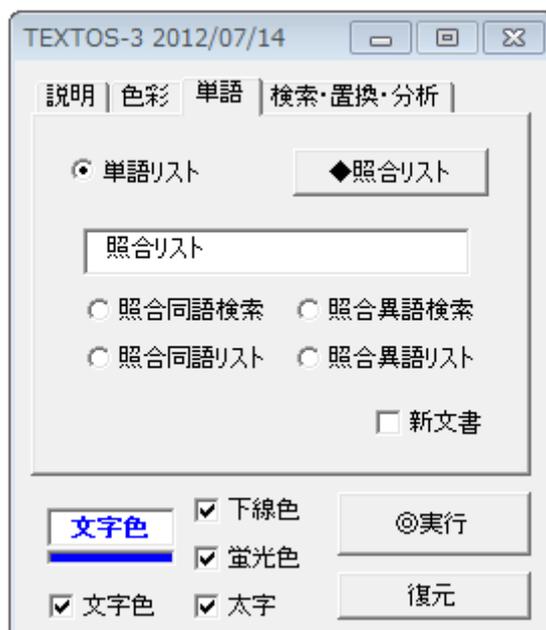
結果：正規表現 ¥z*¥h+ (漢字とひらがな：±漢字+ひらがな)

『{*坊っちゃん*}』夏目漱石

{*親譲りの*}{*無鉄砲で*}{*小供の*}{*時から*}{*損ばかりしている*}。{*小学校に*}{*居る*}{*時分学校の*}{*二階から*}{*飛び*}{*降りて*}{*一週間ほど*}{*腰を*}{*抜かした*}{*事がある*}。{*なぜそんな*}{*無闇をしたと*}{*聞く*}{*人があるかも*}{*知れぬ*}。{*別段深い*}{*理由でもない*}。{*新築の*}{*二階から*}{*首を*}{*出していたら*}、{*同級生の*}{*一人が*}{*冗談に*}、{*いくら*}{*威張っても*}、{*そこから*}{*飛び*}{*降りる*}{*事は*}{*出来まい*}。{*弱虫や*}{*一い*}。{*と*}{*囃したからである*}。{*小使に*}{*負ぶさって*}{*帰って*}{*来た*}時、{*おやじが*}{*大きな*}{*眼をして*}{*二階ぐらいから*}{*飛び*}{*降りて*}{*腰を*}{*抜かす*}{*奴があるかと*}{*云ったから*}、{*この*}{*次は*}{*抜かさずに*}{*飛んで*}{*見せますと*}{*答えた*}。

5. 単語

Word のテキスト内にある単語を抽出し、異なり語のリストを作成するプログラムです。また Word のテキスト内の単語（または単語リスト）を、先に作成した単語リストと照合して、その異同を確かめます。



opt 単語リスト	テキストまたはリストにある単語（異なり語）のリストを作成します。
cmd 照合リスト	次の同語検索と異語検索に使う照合文書を入力します。
opt 照合同語検索	現文書の単語が照合リストに存在すれば、それを青色でマークします。あらかじめ、文書の中でよく使われる語のリストを作成すれば、繰り返された語をチェックすることができます。また、注目すべき語をマークすることができます。
opt 照合異語検索	現文書の単語が照合する単語リストに存在しなければ、それを緑色でマークします。スペルチェックに利用します。
opt 照合同語リスト	照合リストに存在する現文書の単語リストを出力します。
opt 照合異語リスト	照合リストに存在しない現文書の単語リストを出力します。
chk 新文書	新文書に出力します。
cmd 実行	選択したオプションを実行します。

データ例(1)

単語リスト：テキストの異なる単語リストを作成します。

5.1. 単語リスト

単語
リスト
：
テキスト
の
異なる
を
作成
します
。

5.2. 照合同語検索

上の単語リストの出力を単語.docxとして保存し、これを照合リストとします。「照合リスト」ボタンをクリックして、この単語.docxを選択してください。これは変更しない限り一度だけです。

データ例(2)

異なる単語の頻度を計算します。

照合異語検索

異なる単語の頻度を計算します。

5.3. 照合同語リスト

異なる
単語
の
を
します
。

5.4. 照合異語リスト

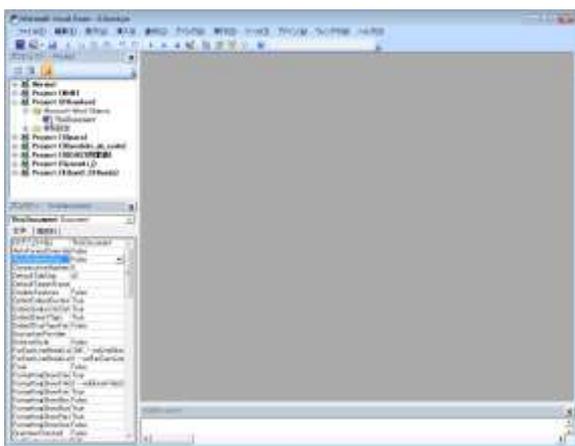
頻度 計算

6. Word VBA プログラミング

6.1. コード

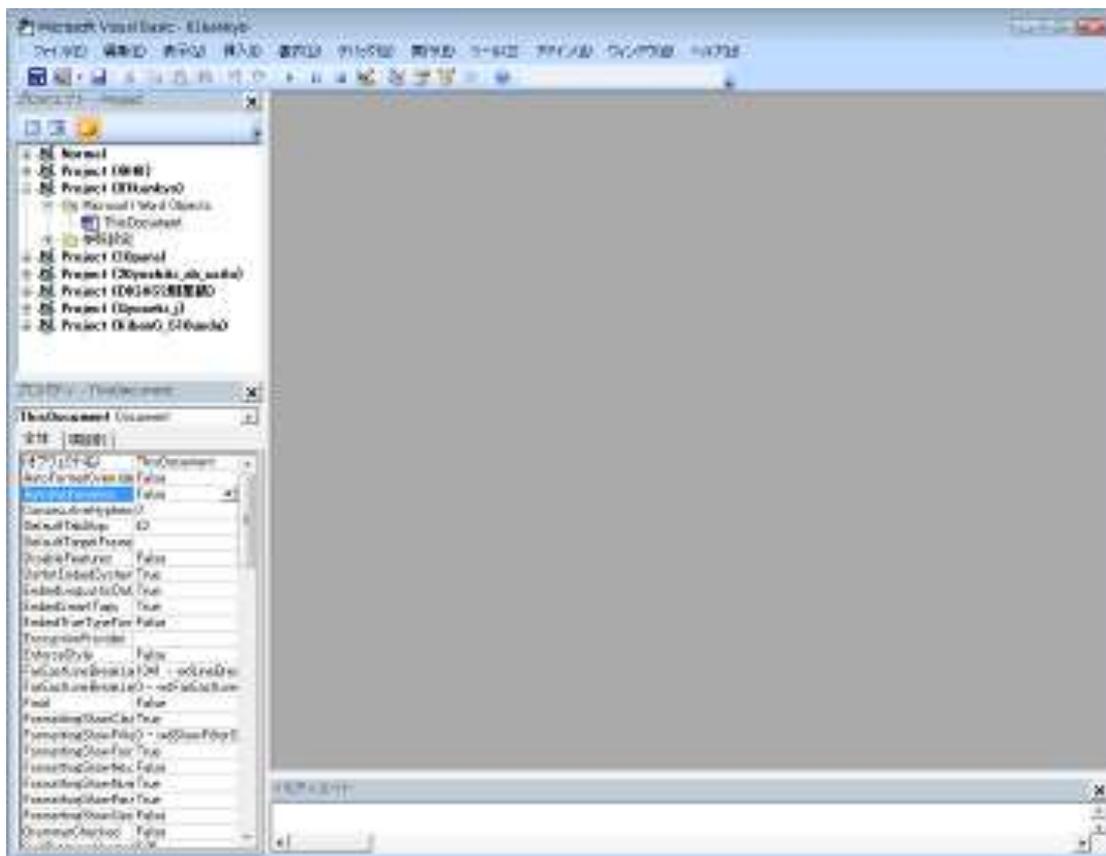
プログラムは一連のコードによって作動します。はじめは簡単なコードの練習をしましょう。

* 「開発」 → Visual Basic, またはショートカット [Alt]+F11



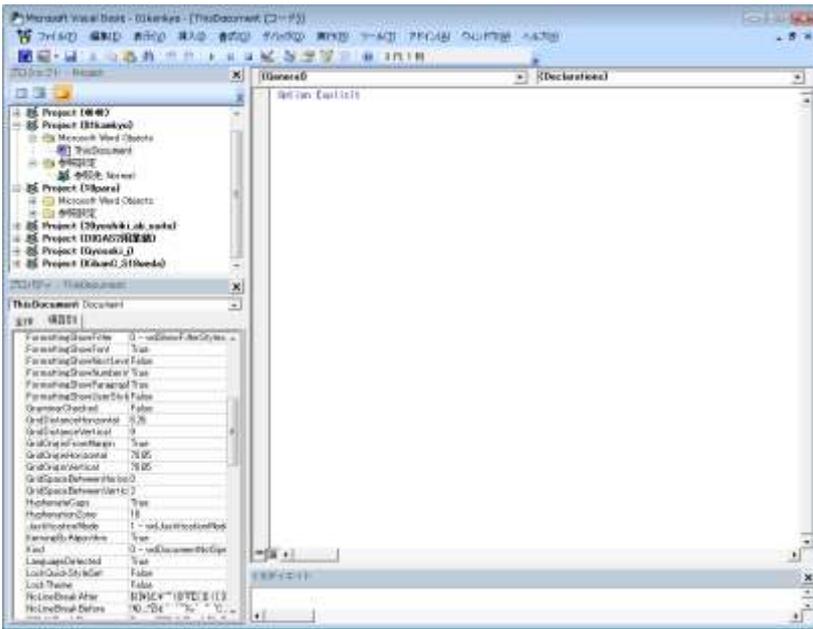
はじめに全体の画面の構成を見ましょう。次の要素を確認しましょう。

画面全体の構成

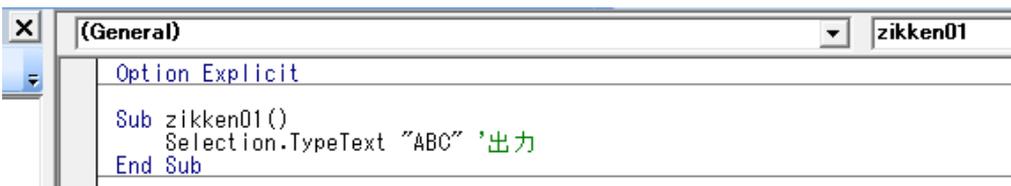


タイトルバー	一番上の青い部分です。ここを選択してドラッグすることができます。最小化、最大化（元に戻す・縮小）、閉じるのボタンがあります。
メニューバー	ファイルからヘルプまでのメニューが並んでいます。
ツールバー	さまざまな機能のショートカットとして使用します。
プロジェクトエクスプローラ	左上のペインでファイルやユーザーフォーム、標準モジュールなどが並びます。
プロパティウインドー	左下のペインで、現在選択されているオブジェクトのプロパティを表示・設定します。
イミディエイトウインドー	プログラムの動作中にストップして、変数の動きを見るときに使用します。表示されていないときは、*「表示」→「イミディエイト・ウインドー」

*新しい文書を開き、[Alt]+F11 で **Visual Basic Editor** を立ち上げ、「挿入」→「標準モジュール」を開きます。



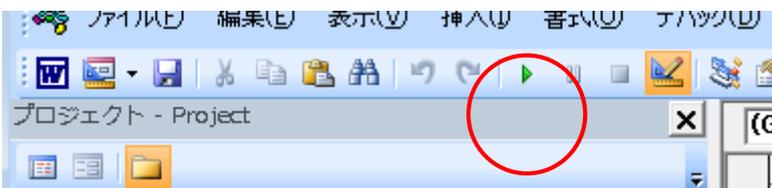
* 次のコードを入力します。



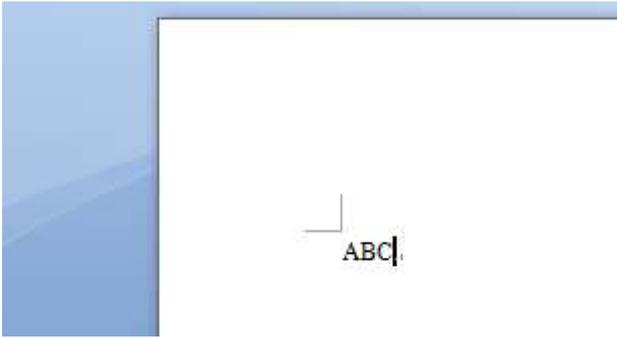
コード：

```
Sub zikken01()
    Selection.TypeText "ABC" '出力
End Sub
```

* このコードを実行するには、ツールバーの「実行」ボタンを押します。ショートカットは[F5]です。



次がその結果です。

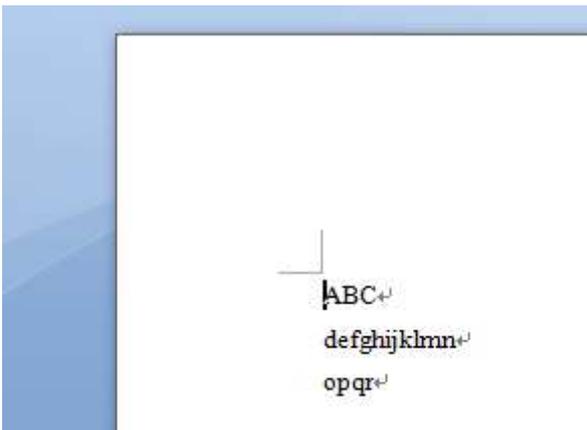


6.2. 繰り返しの制御文

「繰り返しの制御文」を使って段落に番号をつけるためのプログラムを作ります。

(1) 次のような 3 行を選択して新文書に貼り付けます。

```
ABC
defghijklmn
opqr
```



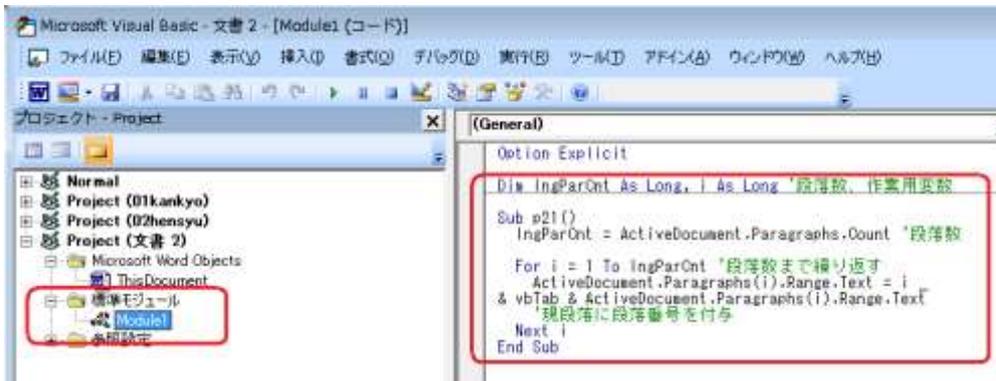
(2) Visual Basic Editor(VBE) の標準モジュールに次のコードを記入します。

```
Sub p21()
    Dim Pc&, i& '段落数、作業用変数

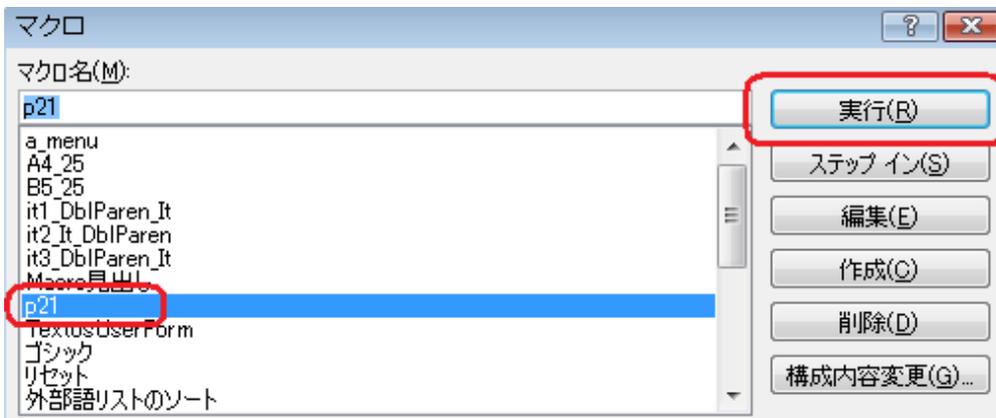
    Pc = ActiveDocument.Paragraphs.Count '段落数

    For i = 1 To Pc '段落数まで繰り返す
        ActiveDocument.Paragraphs(i).Range.Text = _
            i & vbTab & ActiveDocument.Paragraphs(i).Range.Text
        '現段落に段落番号を付与
    Next i
```

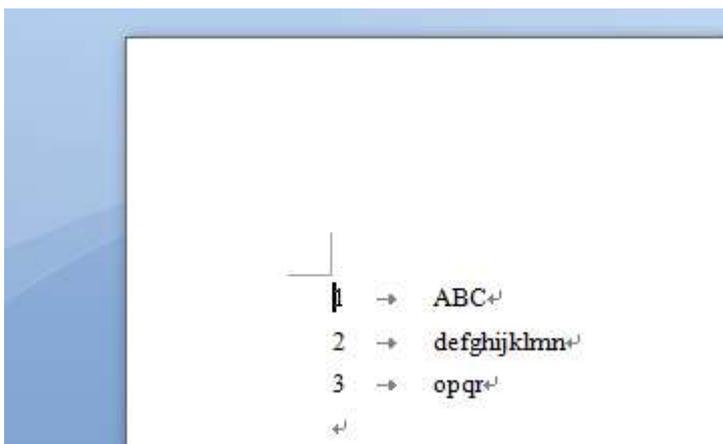
End Sub



(3) 文書に戻って、[Alt]+F8, 「マクロ名」 から p21 を選択し、実行します。



《結果》



二重のネスト

連続する日本語文を WordVBA の Words(*Index*)コレクションを用いて単語に分割します。

```

Sub p22() '日本語文を語に分割
    Dim Pc&, Dn$, Wc&, i&, j& '段落数、文書名、語、作業用変数

    Pc = ActiveDocument.Paragraphs.Count '段落数
    Dn = ActiveDocument.Name '現文書名
    Documents.Add '新文書を加える

    For i = 1 To Pc '段落数まで繰り返す
        Wc = Documents(Dn).Paragraphs(i).Range.Words.Count
        '現段落の語数

        For j = 1 To Wc '語数まで繰り返す
            Selection.TypeText _
                Documents(Dn).Paragraphs(i).Range.Words(j).Text & "/"
            'を付与
        Next
    Next
End Sub

```

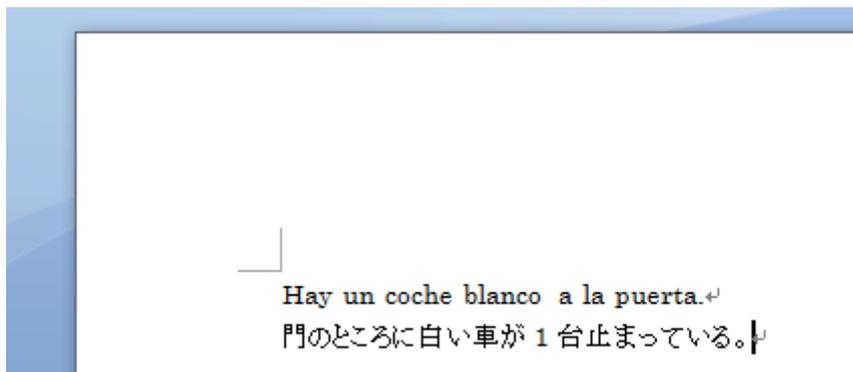
実行：

次の文を新文書に書き込み、p22 を実行します。

```

Hay un coche blanco a la puerta.
門のところに白い車が 1 台止まっている。

```



結果：

```

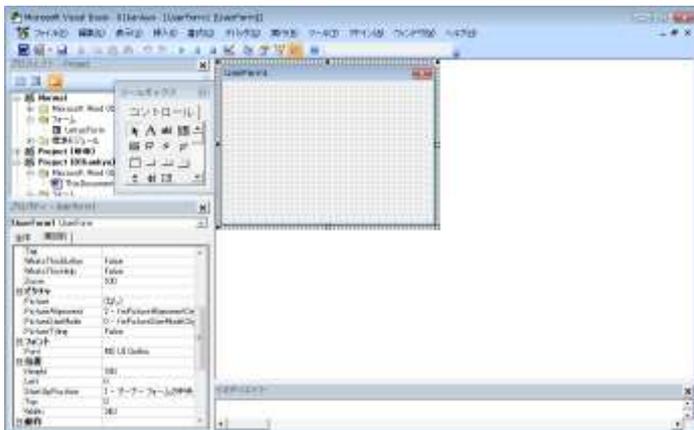
Hay /un /coche /blanco /a /la /puerta./
/門/の/と/こ/ろ/に/白/い/車/が/1/台/止/ま/っ/て/い/る/。/

```

6.3. ユーザーフォーム

コードだけでもプログラムを実行できますが、ユーザーフォームにコードを結び付けておくと、さらにプログラムが使いやすくなります。

* 「挿入」 → 「ユーザーフォーム」



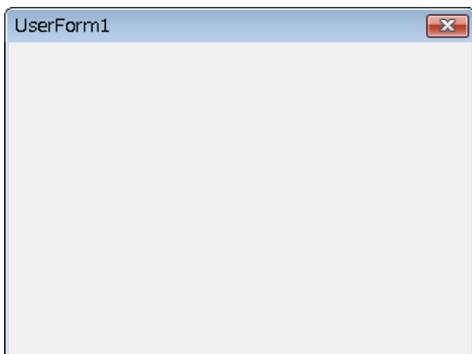
コード画面に戻るには、エクスプローラの「コードの表示」ボタン、または F7。再びオブジェクト画面に戻るには、エクスプローラの「オブジェクトの表示」ボタン、または[Shift]+F7。

ユーザーフォームの起動させるには、標準モジュールに次を書き込みます。

コード：

```
Sub UserFormShow() 'ユーザーフォーム
    UserForm1.Show 'ユーザーフォームを立ち上げる
End Sub
```

《実行》



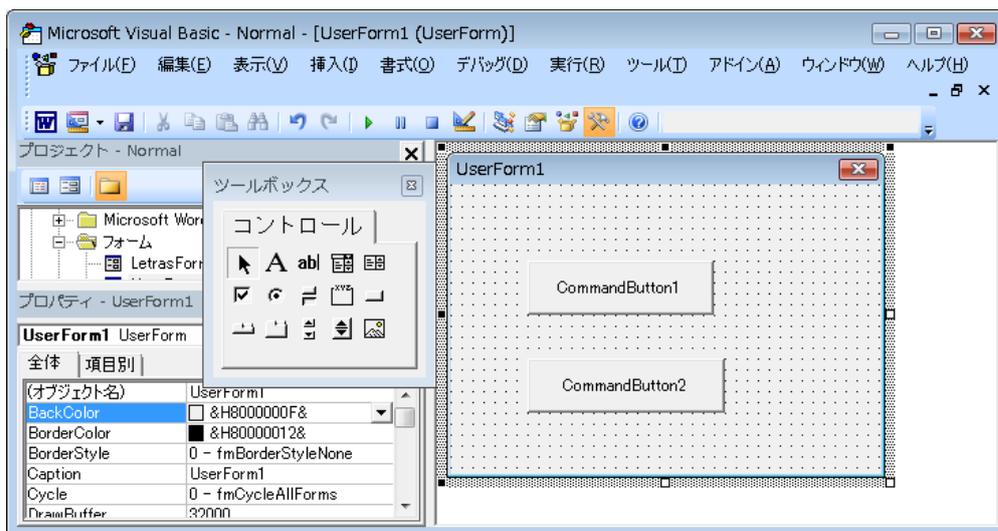
ツールボックス

次の「ツールボックス」を表示します。

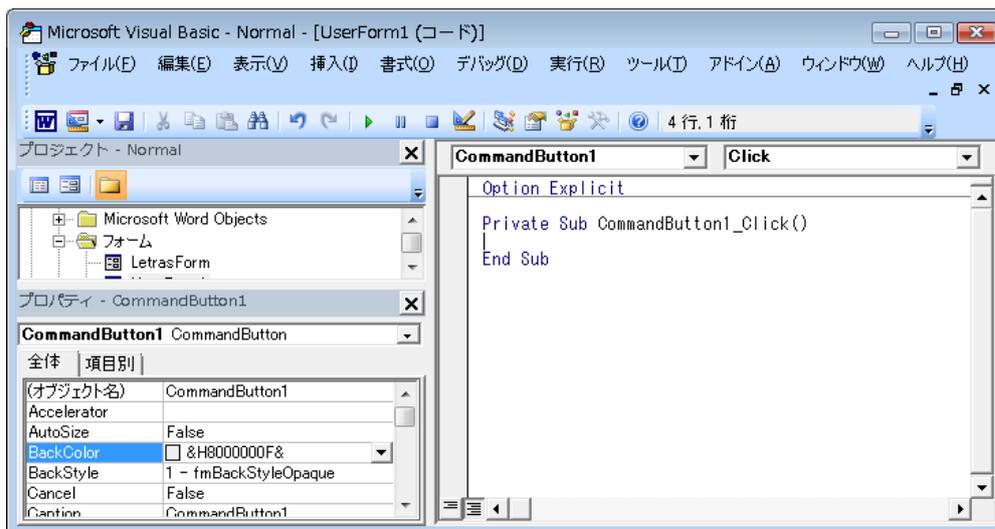


コントロールには、「ラベル」「テキストボックス」「コンボボックス」「リストボックス」「チェックボタン」「オプションボタン」「トグルボタン」「フレーム」「コマンドボタン」「タブストリップ」「マルチページ」「スクロールバー」「スピンドボタン」「イメージ」があります。

* 次のように 2 つのコマンドボタンを配置します。



* CommandButton1 をダブルクリックすると、次のコード画面が現れます。



*カーソルの位置に次のコードを書き込みましょう。

コード：

```
Selection.TypeText "ABC" '出力
```

*次を実行します。F5, F5, F5, ストップ（右上の X 印）

結果：

```
ABCABCABC
```

標準モジュールにコードを記入します。

CommandButton2 をダブルクリックし、カーソルの位置に次のコードを入れます。

コード：

```
End
```

6.4. マクロの記録

「マクロの記録」の作業を通して文書を装飾するためのコードを調べましょう。はじめに次のような文を新文書に書き込みます。

```
Hay un coche blanco a la puerta.
```

次を実行しましょう。

* 「開発」 → 「マクロの記録」

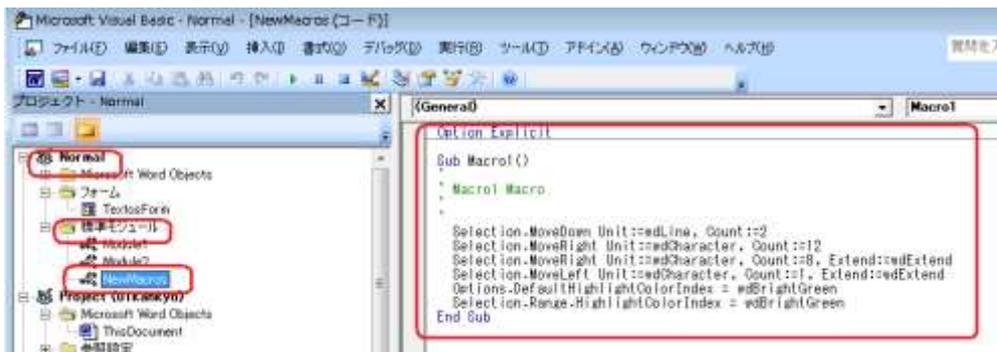


* 「blanco」を緑色の蛍光ペンでマークします。

Hay un coche **blanco** a la puerta.
 門のところに白い車が1台止まっている。

* 「開発」→「記録終了」

この操作のコードは VBE のプロジェクトエクスプローラの Normal の標準モジュールの New Macros の Macro1 に記録されています。



```

Sub p23()
    '
    ' Macro1 Macro
    '
    '
    Selection.MoveRight Unit:=wdCharacter, Count:=6, Extend:=wdExtend
  
```

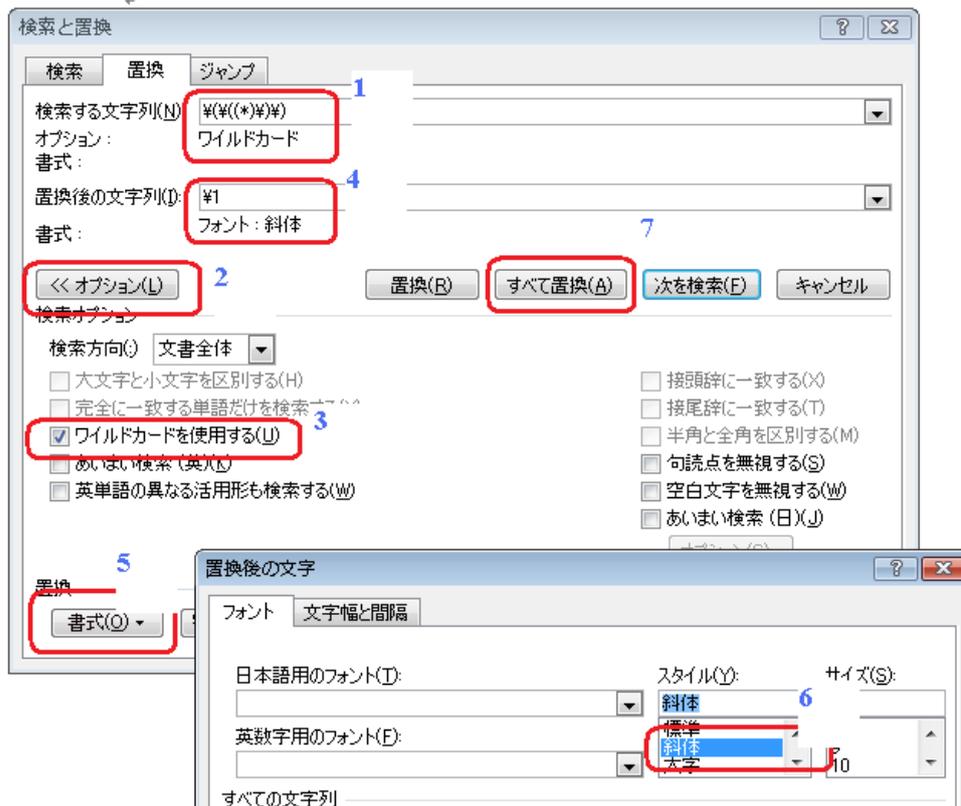
```
Options.DefaultHighlightColorIndex = wdBrightGreen
```

```
Selection.Range.HighlightColorIndex = wdBrightGreen
```

```
End Sub
```

次に、一定の記号で囲まれた文字列の装飾を「置換」の機能を用いて変更します。

Hay un coche ((blanco)) a la puerta.



1. 検索する文字列に、¥¥((*)¥¥¥)を記入します。
2. 置換後の文字列に¥1 を記入します。
3. オプションボタンをおし、オプションの画面を開きます。
4. ワイルドカードを使用する、をチェックします。
5. 「書式」ボタンを押し、「置換後の文字」の「フォント」のスタイルを「斜体」にします。

結果：

Hay un coche *blanco* a la puerta.

この動作のコードをマクロの記録によって調べましょう。

```
Sub p23b() '文字装飾
```

```

ActiveDocument.Content.Copy '現文書をコピー
Documents.Add.Content.Paste '新文書を加えてペースト
With ActiveDocument.Content.Find '現文書の内容を検索
    .ClearFormatting 'フォーマットを初期化
    .Replacement.ClearFormatting '置換フォーマットを初期化
.Text = "¥(¥(*)¥)¥" '検索文字列
    .Replacement.Text = "¥1" '置換文字列
    .Replacement.Font.Italic = True
    .Forward = True
    .Wrap = wdFindStop
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchByte = False
    .MatchAllWordForms = False
    .MatchSoundsLike = False
    .MatchFuzzy = False
    .MatchWildcards = True
.Execute Replace:=wdReplaceAll '全部変換
    End With
End Sub

```

Word の検索フォームのオプションは、次のコードに対応します。

MatchCase	大文字と小文字を区別する(H)
MatchWholeWord	完全に一致する単語だけを検索する(Y)
MatchWilCards	ワイルドカードを使用する(U)
MatchSoundsLike	あいまい検索 (英) (K)
MatchAllWordForms	英単語の異なる活用形も検索する(W)
MatchByte	半角と全角を区別する(M)
MatchFuzzy	あいまい検索 (日) (J)

* なお、MatchWilCards, MatchSoundsLike, MatchAllWordForms, MatchFuzzy は 1 つだけしか使うことができません。

検索と置換

この Word の動作を利用して、次のようなプログラムを作成しましょう。

```

Sub p31WildCard() 'ワイルドカードによる検索
  Dim strObj As String '現文書のテキスト
  Dim strSch As String 'インプットボックス

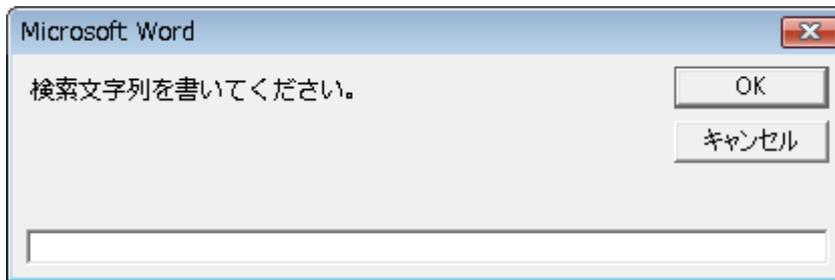
  Options.DefaultHighlightColorIndex = wdBrightGreen
  'ハイライトの色を明るい緑とする

  strSch = InputBox("検索文字列を書いてください。")
  'インプットボックス
  If strSch = "" Then End 'キャンセル

  With Selection.Find '選択範囲で検索
    .ClearFormatting: .Replacement.ClearFormatting
    '検索・置換文字列の書式を削除
    .Text = strSch '検索文字列
    .Replacement.Text = "" '置換文字列
    .Format = True: .Replacement.Highlight = True
    '書式を含む：ハイライト
    .Forward = True: .Wrap = wdFindStop
    '検索を下方向に：末尾で検索終了
    .MatchCase = True: .MatchByte = True
    '大・小文字を区別：全・半角文字を区別
    .MatchWholeWord = False '単語の一部も検索
    .MatchWildcards = False 'ワイルドカード使用
    .MatchAllWordForms = False '活用形検索をしない
    .MatchSoundsLike = False '類似単語検索をしない
    .MatchFuzzy = False '曖昧検索をしない
    .Execute Replace:=wdReplaceAll '全置換
    .ClearFormatting '検索文字列の書式を削除
    .Replacement.ClearFormatting
    '置換文字列の書式を削除
  End With
End Sub

```

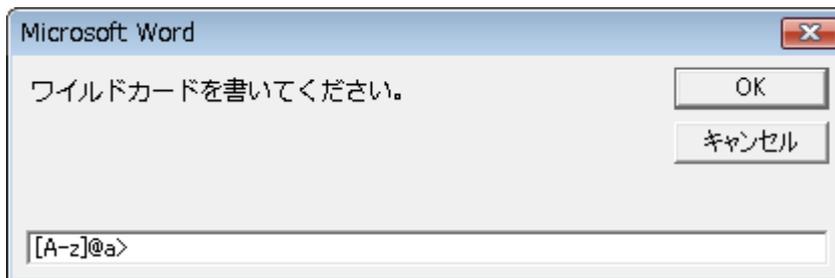
実行 :



結果 :

(1:0) Cap. 1 (1:1) In principio creavit Deus caelum & terram. (1:2) Terra autem erat inanis & vacua: & tenebrae erant super faciem abyssi: & spiritus Dei ferebatur super aquas. (1:3) Dixitque Deus. Fiat lux. Et facta est lux. (1:4) Et vidit Deus lucem quod esset bona: & divisit lucem a tenebris:

実行 :



結果 :

(1:0) Cap. 1 (1:1) In principio creavit Deus caelum & terram. (1:2) Terra autem erat inanis & vacua: & tenebrae erant super faciem abyssi: & spiritus Dei ferebatur super aquas. (1:3) Dixitque Deus. Fiat lux. Et facta est lux. (1:4) Et vidit Deus lucem quod esset bona: & divisit lucem a tenebris:

6.5. VBScript オブジェクト

(a) 連想配列オブジェクト

Word VBA, Excel VBA に VBScript の「ディクショナリ」 Scripting.Dictionary を生成して連想配列を使うことができます。

```
Sub UserForm_Activate() 'ユーザーフォームを表示
```

```

Dim objDic As Object '連想配列オブジェクト
Set objDic = CreateObject("Scripting.Dictionary")
'連想配列オブジェクトを生成
End Sub

```

一般の配列では、配列のキーを 0 と正の整数を使いますが、連想配列では文字列を使うことができます。（配列のアイテムは文字でも数値でもかまいません。）たとえば、上のコードのように objDic を生成すると、objDic("ABC") =123 のように文字列 ABC をキーとして、その位置にアイテム 123 を代入することができます。

言語データ分析では文字列の処理が中心となりますので、それをキーにすることができればとても便利です。たとえば、連想配列にキーが存在しているか否かは Exists という関数を使います。これによって、リストにある文字列の有無を検索（サーチ）することなく、一瞬で判別でき、また、キーに関する情報をアイテムとして代入しておけば、その取得も一瞬で可能になります。

(b) 正規表現オブジェクト

Word VBA, Excel VBA に VBScript の「正規表現」VBScript.RegExp を生成すれば、正規表現による検索・置換ができます。

```

Option Explicit

Dim obj 正規表現 As Object

Sub 正規表現()
Set obj 正規表現 = CreateObject("VBScript.RegExp")
'正規表現オブジェクト生成
obj 正規表現.Global = True '全体検索

MsgBox 正規表現置換$("{*}", "([{}]<>¥[¥])", "¥$1")
'括弧類に¥をつけてエスケープ

Set obj 正規表現 = Nothing '正規表現オブジェクト解放
End Sub

Private Function 正規表現置換$(ByVal 対象$, 検索$, 置換$)
obj 正規表現.Pattern = 検索$ '検索パターン
正規表現置換$ = obj 正規表現.Replace(対象$, 置換$) '置換
End Function

```

上のように、正規表現置換の関数を用意しておくこと、VBA の `Replace` 関数のように、3つの引数で置換ができます。

6.6. 説明



フォームを立ち上げるために、標準モジュールに次のコードを用意します。

Option Explicit

```
Declare Function FindWindow& Lib "user32.dll" Alias "FindWindowA" _  
(ByVal C$, ByVal w$) 'ハンドル
```

```
Declare Function GetWindowLong& Lib "user32.dll" Alias "GetWindowLongA" _  
(ByVal F&, ByVal i&) 'ボタン制御
```

```
Declare Function SetWindowLong& Lib "user32.dll" Alias "SetWindowLongA" _  
(ByVal F&, ByVal i&, ByVal N&) 'メニューバー
```

```
Declare Function SetWindowPos& Lib "user32.dll" (ByVal HW&, ByVal Ins&, _  
ByVal x&, ByVal y&, ByVal C&, ByVal D&, ByVal F&) 'フォーム表示状態
```

```
Dim F&, G&, S& '最小化ボタン用変数
```

最小化ボタンを使うために必要な関数と変数です。

```
Sub Textos()
```

```
Call フォーム表示(fmTextos)
End Sub
```

基本的に上のコードでフォーム(fmTextos)が立ち上がります。これは次のサブルーチンを呼び出します。

```
Sub フォーム表示(Frm)
    Frm.Show vbModeless 'フォームをモードレスで表示

    F& = FindWindow("ThunderDFrame", Frm.Caption) 'ハンドル
    G& = GetWindowLong(F&, -16) Or &H20000 'ウィンドウの情報
    S& = SetWindowLong(F&, -16, G&) 'フォームボタン

    ●フォーム前面
End Sub
```

```
Public Sub ●フォーム前面()
    S& = SetWindowPos(F&, -1, 150, 150, 0, 0, 1) '表示 (ハンドル, 前面, x, y, 0, 0, 1)
End Sub
```

フォームを最前面に表示するためのコードです。

```
Public Sub ●フォーム後面()
    S& = SetWindowPos(F&, 1, 150, 150, 0, 0, 1) '表示 (ハンドル, 後面, x, y, 0, 0, 1)
End Sub
```

フォームを後面に表示するためのコードです。

このフォームの最初の「説明」のタブのページには次のコマンドとオプションがあります。

オブジェクト名	Caption または Value
pg 説明	説明
lbl 説明	TEXTOS-2 WordVBA による...
chk 文字色	文字色
chk 下線色	下線色
chk 蛍光色	蛍光色
chk 太字	太字

txt 文字色	文字色（前図では青色）
txt 下線色	前図の青線
cmd 実行	◎実行
cmd 消去	消去
cmd 復元	復元

このページでは下部にあるすべてのチェックボックス(chk)とコマンドボタン(cmd)は可動しません(Enabled=False)。

Option Explicit '変数を明示

Private Declare Function ColorHLSToRGB& Lib "SHLWAPI.DLL" _
(ByVal H%, ByVal L%, ByVal S%) 'RGB 関数

Private Declare Sub ColorRGBToHLS Lib "SHLWAPI.DLL" _
(ByVal clrRGB&, H%, L%, S%) 'HLS サブルーチン

Dim obj 照合配列 As Object, obj 連想配列 As Object

Dim obj 正規表現 As Object, obj 正規関数 As Object

Dim 一致 v, 一致コレクション v

Dim obj エクセル As Object, obj クリップボード As New DataObject

Dim lngRGB&, HexRGB\$ '長整数型 RGB 値、16 進数 RGB 値

Dim 蛍光色配列\$(), 開始時間 v, i%

Dim 出力文字数&, 出力文字列\$, 追加文字列\$

Dim 単語配列\$(), 単語 v

Dim 選択範囲\$, 段落配列\$(), 段落 v, 段落番号&

Dim 検索置換式\$(), 検索\$(), 置換\$()

これらの関数や変数については以下で（「色彩」「単語リスト」「検索置換」「分析」）説明します。連想配列オブジェクトと正規表現オブジェクトについては→「準備」

Sub UserForm_Activate() 'フォーム表示

Set obj 照合配列 = CreateObject("Scripting.Dictionary")

Set obj 連想配列 = CreateObject("Scripting.Dictionary")

Set obj 正規表現 = CreateObject("VBScript.RegExp")

obj 正規表現.Global = True '全体検索

obj 正規表現.MultiLine = True '^が行頭に、\$が行末に一致

```

Set obj 正規関数 = CreateObject("VBScript.RegExp")
obj 正規関数.Global = True '全体検索
obj 正規関数.MultiLine = True '^が行頭に、$が行末に一致

Set obj エクセル = CreateObject("Excel.Application")

蛍光色配列$ = Split("FFFFFF,000000,FF0000,FFFF00,00FF00,FF00FF," _
    & "0000FF,00FFFF,FFFFFF,800000,808000,008000,800080,000080," _
    & "008080,808080,C0C0C0", ",")

HexRGB$ = txt 文字色値

cbo 蛍光色.AddItem "蛍光色":    cbo 蛍光色.AddItem "黒"
cbo 蛍光色.AddItem "青":        cbo 蛍光色.AddItem "水色"
cbo 蛍光色.AddItem "明るい緑":  cbo 蛍光色.AddItem "ピンク"
cbo 蛍光色.AddItem "赤":        cbo 蛍光色.AddItem "黄"
cbo 蛍光色.AddItem "白":        cbo 蛍光色.AddItem "濃い青"
cbo 蛍光色.AddItem "青緑":      cbo 蛍光色.AddItem "緑"
cbo 蛍光色.AddItem "紫":        cbo 蛍光色.AddItem "濃い赤"
cbo 蛍光色.AddItem "濃い黄":    cbo 蛍光色.AddItem "50%灰色"
cbo 蛍光色.AddItem "25%灰色":   cbo 蛍光色.ListIndex = 0 '最初の項目を選択

cbo 検索.AddItem "リテラル検索置換":  cbo 検索.AddItem "ワイルド検索置換"
cbo 検索.AddItem "単純検索置換":      cbo 検索.AddItem "正規表現検索置換"
cbo 検索.AddItem "検索式頻度":        cbo 検索.AddItem "鍵語頻度"
cbo 検索.AddItem "鍵語外置":          cbo 検索.AddItem "鍵語内置"
cbo 検索.ListIndex = 0 '最初の項目を選択

ページ.Value = 0 '「説明」のタブを選択
End Sub

```

以下で使用するオブジェクト（照合配列、連想配列、正規表現、正規関数、エクセル）をセットし、蛍光色配列\$と HexRGB\$の初期値を代入し、cbo 蛍光色(→「色彩」)のコンボボックスの要素名を設定します。

```
Select Case ページ.Value 'ページでケースを選択
```

```

Case 0 '説明
chk 文字色.Enabled = False: chk 蛍光色.Enabled = False
chk 下線色.Enabled = False: chk 太字.Enabled = False
cmd 実行.Enabled = False: cmd 復元.Enabled = False

Case 1 To 3 '色彩、単語リスト、検索置換
chk 文字色.Enabled = True: chk 蛍光色.Enabled = True
chk 下線色.Enabled = True: chk 太字.Enabled = True
cmd 実行.Enabled = True: cmd 復元.Enabled = True

If cbo 検索.ListIndex >= 4 And cbo 検索.ListIndex <= 7 Then
chk 文字色.Enabled = False: chk 蛍光色.Enabled = False
chk 下線色.Enabled = False: chk 太字.Enabled = False
cmd 実行.Enabled = True: cmd 復元.Enabled = True
End If
End Select
End Sub

```

フォームのページの選択にしたがってフォームの下部にある各種のチェックボタンとコマンドボタンの可動性を設定します。

```

Sub cmd 実行_Click(): On Error GoTo Errores 'エラー処理
開始時間 v = Timer: cmd 実行.Caption = "○実行中": DoEvents

If Len(Selection) < 2 Then
If MsgBox("全範囲を選択しますか?", vbYesNoCancel, "範囲選択") = vbYes Then
Selection.WholeStory '全範囲選択
Else
GoTo FIN '実行終了
End If
End If

出力文字列$ = "": 出力文字数& = 0 '初期化

Select Case ページ.Value 'ページでケースを選択
Case 1: ■ 色彩
Case 2: ■ 単語

```

Case 3: ■検索置換分析

End Select

FIN:

cmd 実行.Caption = "●実行:" & Int(Timer - 開始時間 v) & "秒"

Exit Sub

Errores:

●フォーム後面: MsgBox "●エラー:" & Error(Err): ●フォーム前面

End Sub

実行ボタンクリック時に、「実行準備」サブルーチンで範囲の指定を確認し、出力文字に関する変数を初期化します。次に、それぞれのオプションボタンの選択に従って対応するサブルーチンを呼び込みます。

Sub cmd 復元_Click(): On Error GoTo Errores 'エラー処理

Select Case ページ.Value

Case 1: ActiveDocument.Undo '色彩→元に戻す

Case 2: '単語

If chk 新文書 Then

ActiveDocument.Close SaveChanges:=wdDoNotSaveChanges 'セーブせずに閉じる

Else

ActiveDocument.Undo '元に戻す

End If

Case 3: '検索・置換・分析

If cbo 検索.ListIndex <= 3 Then

ActiveDocument.Undo '元に戻す

Else

obj エクセル.ActiveWorkbook.Close SaveChanges:=False 'ブックをセーブせず閉じる

End If

End Select

Exit Sub

Errores:

```
●フォーム後面: MsgBox "●エラー: " & Error(Err): ●フォーム前面  
End Sub
```

開いた新文書（単語リストまたは検索置換）やエクセルシート（「分析」）をセーブしないで閉じます。ActiveDocument.Undo '元に戻す、はショートカット Ctrl+Z と同じ動作をします。

```
Sub UserForm_QueryClose(Cancel%, CloseMode%) '×終了ボタン  
Set obj 照合配列 = Nothing: Set obj 連想配列 = Nothing  
Set obj 正規表現 = Nothing: Set obj 正規関数 = Nothing  
Set obj エクセル = Nothing  
End '終了  
End Sub
```

×印終了ボタンを押したときに連想配列を開放し終了します。

6.7. 参考書

(a) Word VBA に関する参考書

*Excel VBA については数多くの参考書がありますが、Word VBA はそれが限られています。そのなかから現在入手できる次の参考書を選びました。

Hart-Davis, Guy. (2009) *Word 2007 Macros & VBA Made Easy*, New York, McGraw-Hill. (入門者向けの丁寧な解説です。)

西上原裕明 (2006)『Word で実践、やさしくて役立つ「マクロ」事例集』東京、技術評論社（この本から勉強を始めることを勧めます。)

西上原裕明 (2010)『Word のマクロ実践サンプル集』東京、技術評論社（参考になるプログラム集です。先行のプログラムを読むことで読解と応用力を養いましょう。)

荻野綱雄・田野村忠温 (2011)『講座 IT と日本語研究：アプリケーションソフトの応用』東京、明治書院（入門者には第二章「Word の応用」（谷本玲大）が役立ちます。)

Roman, Steven. (1998) *Learning Word Programming*, Sebastopol, CA, O'Reilly （Word VBA を俯瞰するための本格的な参考書です。英語の参考書を使って用語に慣れておくとウェブの情報が得られやすくなります。)

Savikas, Andrew. 日向あおい訳 (2005)『Word Hacks プロが教える文書活用テクニック』東京、オライリー・ジャパン（高度なテクニックを扱う上級者向けの参考書

です。)

土屋和人 (2013)『Word マクロ/VBA 徹底入門』(Word2013 を含めた最新の情報が載せられています。これからプログラミングを始める人は p.93 までの「基礎知識」をしっかりと理解してください。)

*書籍以外でもウェブで「Word VBA *** (質問事項)」(たとえば Word VBA Paragraphs)を検索すると多くの疑問点を解決することができます。そのときプログラムのコードが読めることが必須になります。授業や書籍などを参考にして、コードの読み方と書き方を習得してください。また、Excel VBA と共通する多数の操作や関数があります。

(b) VBScript に関する参考書

*Word と Excel の VBA の中で VBScript というスクリプト言語の「連想配列ディクショナリ」オブジェクトと検索・置換のための「正規表現」オブジェクトを使用することができます。VBScript については次の参考書が有用です。

アंक (株) (1999)『改訂版 VBScript ポケットリファレンス』東京、技術評論社 (項目ごとに簡単な用例をつけて使用法が解説されています)

井川はるき (2006)『VBScript 逆引き大全』東京、秀和システム (項目ごとにさらに高度な使用法が説明されています)

佐藤信正 (2006)『VBScript 実用プログラミングテクニック』メディア・テック出版 (正規表現の詳しい説明があります)

結城圭介 (2010)『VBScript サンプル大全集』東京、技術評論社 (応用法が参考になります。)