

1 テキストエディタとは

コンピュータで文章を作成、保存するためのファイル形式にはいくつかのものがああります。例えば、MS Word のファイル形式は、拡張子が .doc (MS Word 2003 以前) か、.docx (MS Word 2007 以降) となっている Word 文書です。また、Windows に付属のメモ帳のファイル形式は、拡張子が .txt となっているテキストファイルです。

テキストエディタとは、このテキストファイルを作成、編集するためのアプリケーションの総称です。上述のメモ帳もテキストエディタの一種なのですが、メモ帳以外にも数多くのテキストエディタが存在します。この文書では、テキストエディタの使い方を説明します。

まず、そもそもテキストファイルとはどういうものなのでしょうか。コンピュータ上では、文字はすべて文字コードで記録されています。つまり、文字毎に数字が割り当てられていて、ファイルの中で文字はその数字に置き換えて記録されています。この文字コードをファイルとしてそのまま保存したものが、テキストファイルです。つまり、文字だけで書かれたファイルがテキストファイルです。

しかし、文字コードには、文字に付随する情報、例えば、「この文字を太字にする」とか、「この文字の色は赤にする」とかいった情報は含まれていません。ですので、テキストファイルにそのような情報をそのまま含めることはできません。MS Word のようなワープロソフトは、それぞれ独自の形式でそのような情報を保存しています。また、ワープロソフトの文書では、図表の情報も独自の形式で保存されています。ですので、Word 文書には、文字の装飾や図表なども含めることができます。この独自の形式というのは、文字で書かれたようなものではありません。ですので、Word 文書はテキストファイルではありません。このようなテキストファイルではないファイルをまとめてバイナリファイルと呼びます。Word 文書はバイナリファイルです。同様に、Excel 文書もバイナリファイルです。

テキストエディタとは、このテキストファイルを扱うためのアプリケーションなのです。バイナリファイルは、通常、テキストエディタで扱うことはできません。

なお、上述の通り、Word 文書などのワープロソフトの文書には、文字の装飾や図表を

含めることができますが、テキストファイルはそういうことができません。そういう点では、テキストファイルよりも Word 文書の方が優れているように思えるかもしれませんが、実際は、テキストファイルの方が優れている点もあります。

まず、第一に、テキストファイルは Word 文書よりもサイズが小さくなります。文字の情報や図表などが含まれないので、その分サイズが小さくなります。

また、どのような環境でも閲覧、編集できるということもあります。Word 文書は基本的には MS Word がインストールされていなければ、閲覧、編集ができません。どのコンピュータにも MS Word がインストールされているとは限りません。ただ、実際は、MS Word がなくても、Word 文書を閲覧、編集ができるアプリケーションもあるのですが、そういうものもやはりインストールされているとは限りません。それに対して、テキストエディタはほぼ間違いなくどのコンピュータにもインストールされています。例えば、Windows の場合、テキストエディタとして標準でメモ帳がインストールされています。

さらに、テキストエディタは一般に MS Word と比べて動作が非常に軽快ということがあります。テキストエディタは MS Word と比べて機能が少ないので、動作が軽快になっています。起動速度もだいぶ違います。

そして、文系の研究として重要な点として、テキストファイルは機械的処理が簡単であるという点が挙げられます。文系の研究だと、「複数の文書から特定のパターンのデータを集めて一つのファイルにまとめる」とか「データをアルファベット順に並べ替える」とかいう作業が必要になることがあります。テキスト文書だと、外部のアプリケーションを使うことにより、このような作業もかなり柔軟に、機械的に処理することができます。Word 文書だとこのような作業はあまり柔軟にできません。

なお、Windows の場合、.txt という拡張子の付いたファイルに「テキスト文書」という説明がついています。一般的には、「テキストファイル」と呼ばれるので、この文書では、この呼び方で統一します。また、Web ページを記述するための HTML 文書や表計算ソフトの保存形式である CSV ファイルもテキストファイルです。例えば、HTML 文書は、拡張子が .html か .htm になっていて、テキストファイルとは異なるように見えるかもしれませんが、HTML では、ワープロソフトと同じように、文字の装飾や図表を表現することができます。しかし、HTML では、ワープロソフトの保存形式と異なり、「この文字を太字にする」とか「この文字の色は赤にする」という情報は、HTML タグというものを使って、そのまま文字で表現されています。これを Internet Explorer などの Web ブラウザが解釈して、画面上に表示されることになります。したがって、HTML では、文字の装飾を全てテキストとして定義しているので、テキストファイルになります。ですので、HTML 文書はテキストエディタで開くことができます。同様に CSV ファイルもテキストファイル

なので、テキストエディタで開くことができます。このように拡張子が .txt でなくても、テキストファイルであり、テキストエディタで開くことができるファイル形式も存在します。特定の形式のファイルを扱う場合、それがテキストファイルなのかバイナリファイルなのか意識しなければならないことがあることに注意して下さい。

2 文字コード

この節では、具体的にテキストエディタの使い方を説明する前に文字コードについて説明します。

コンピュータの扱うデータは、コンピュータの内部では、全て数値で表現されています。文書のデータであっても、画像のデータであっても、動画のデータであっても、全てフォーマットごとに決まった形式で数値化されています。もちろん我々は、その数値化の方法を知らなくてもそのデータを扱うことができるわけですが、人文科学、社会科学の研究でコンピュータを使う場合でも、文字データがどのように数値化されているのかということは、非常に役に立つ知識ですので、知っておいて損はありません。

2.1 ASCII

まず、英数字、記号を表現するための文字コードを紹介します。

これは、ASCII (American National Standard Code for Information Interchange) コードと呼ばれるもので、表 1 のように、7 ビット (二進数の 7 桁) を使って英数字、記号、その他制御文字を表します。この文字コードは、その名の通り、もともとアメリカで作られたものですが、現在は、日本においてはもちろんのこと、国際的に使われています。そのため、この ASCII コードを使えば、世界中のどこでもきちんと表示できることが保証されます。

ここで、表 1 の一番上の行の数字が十六進数の上一桁、一番左の列の数字が十六進数の下一桁を表しています。したがって、例えば、大文字の A は、十六進数で 41 という数で表現されます。なお、この数字は十六進数なので、「よんじゅういち」ではなく「よんいち」と読みます。また、一般的に十六進数を数字の前に '0x' を置いて '0x41' と表記することが多いので、以下この文書でも同様に表記します。

この ASCII コード表を眺めてみると、まず、十六進数の 0x00 から 0x7F までだけを使っていることが分かります。0x7F は二進数にすると 1111111 なので、7 ビットだけ使っていることが分かります。

また、この ASCII コード表では、十六進数の 0x20 から 0x7E まで見慣れた英数字と記号が並んでいます。0x20 の SPACE は空白を表しています。さらによく見ると、アルファ

	0	1	2	3	4	5	6	7
0	NUL	DLE	SPACE	0	@	P	'	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	”	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	,	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

表1 ASCIIコード表

ベットの大小文字、小文字が隣合わせでならんでいることが分かります。我々ユーザーにとってはあまり関係ありませんが、これによって、「ASCIIコードに0x20を足したり引いたりすることにより、簡単に大文字小文字の変換ができる」という利点が生れます。

さらに、それ以外の部分、つまり、0x00から0x1Fまでと0x7Fは、制御文字と呼ばれるものです。通常、これらが我々の目に見える形で表示されることはありませんが、ディスプレイやプリンタを制御するための使われる特殊な文字です。それぞれ、ASCIIコード表では、その制御文字の役割を表す二、三文字の記号で表現されています。ここで、制御文字を全て説明することはしませんが、この後出てくるものを三つだけ紹介しておきます。

- LF (line feed) 0x0A: 改行
- CR (carriage return) 0x0D: 復帰
- ESC (Escape) 0x1B: エスケープ

コンピュータの内部では、8ビットをひとまとまりとして扱っているので、256文字

まで同時に扱うことができます。しかし、この ASCII コードは前述の通り 7 ビットなので、その半分しか使っていないこととなります。(表 1 を 8 ビットの表にすると右半分があまります。)この余った部分を使って、ドイツ語やフランス語のウムラウトやアクセントを表す文字コード (ISO-8859-1、もしくは、Latin-1 と呼ばれます。)や、日本語の半角カタカナを表す文字コード (JIS X 0201 と呼ばれます。)が存在します。ただ、インターネット上には、「インターネット上に流れている文字コードは全て 7 ビットだ」ということを前提としているコンピュータが存在するので、メールや Web では、そのままの形で ISO-8859-1 や JIS X 0201 は使ってはならないということになっています。ですので、メールなどでウムラウトやカタカナを表すためには、これとは別の手段を使います。

2.2 日本語の文字コード

前述の ASCII コードは、英数字と記号しか使えません。では、日本語の文字を表現するためにはどうすればいいのでしょうか。

英語の場合、アルファベットの大文字、小文字、数字を合わせても、62 文字しかありません。記号を加えてもそれほど数は多くありません。ですので、7 ビットだけで全ての文字を表すことができます。実際、ASCII コードでは、0x20 から 0x7E までの 95 文字しか使っていません。しかし、日本語の場合、ひらがなだけでも、濁点、半濁点の付いた文字や「ゐ」(わ行の「い」)や「ゑ」(わ行の「え」)も一文字として数えますので、80 文字以上あります。ほぼ、同じ数のカタカナもありますし、漢字は義務教育で学習する常用漢字だけでも 1945 文字あります。とても、7 ビットや 8 ビットでは表現することはできません。

そこで、日本語を表現するために 16 ビット (2 バイト) 使うことになっています。16 ビットだと、2 の 16 乗で 65536 文字使うことができます。これだけあれば、日本語の文字を全て表現できそうです。

実際に日本工業規格 (JIS) の規格として JIS X 0208 と呼ばれる日本語の文字の集合が定められました。この規格には、漢字以外が 524 文字、漢字は使用頻度にあわせて、第一水準漢字が 2,965 文字、第二水準漢字が 3,390 文字の計 6,879 文字が含まれています。この文字の集まりをコンピュータ上で表現する方法として、主に以下の三つの文字コードが使われています。

ISO-2022-JP JIS コードとも呼ばれます。主に、電子メールなどで使われます。文字コードの中に「ここからが JIS X 0208 の文字です。」「ここまでが JIS X 0208 の文字です。」というコードを埋め込むようになっています。これは、「エスケープ」を使って、前者は '0x1B 0x24 0x42' というコードが、後者は '0x1B 0x28 0x42' とい

うコードが使われます。このため、日本語一文字に2バイトを使っていますが、実際には、0x7Fまでしか使用していないので、前述の7ビットの問題は生じません。Shift_JIS「シフトJIS」「Shift-JIS」とも表記されます。主にWindowsやMac OSで使われている文字コードです。日本語の文字をASCIIコードで使っていない0x80以降に組み込んでいるので、7ビットの問題があります。電子メールでは使えません。Windowsでメールを使う時には、メールソフトが自動的にISO-2022-JPに変換しています。

EUC-JP EUCとはExtended UNIX Codeの略です。その名の通り、UNIXやLinuxなどで使われています。Shift_JISと同様に7ビットの問題があります。

なお、現在、日本語の文字の集合としては、前述のJIS X 0208を拡張したJIS X 0213という規格が定められています。これは、JIS X 0208を拡張したもので、JIS X 0208に第三水準漢字1,249文字、第四水準漢字2,436字が追加されています。さらにこのJIS X 0213は2004年に改正され、168文字の字体が変更されました。この改訂版は、一般的には「JIS2004」と呼ばれています。このJIS2004では、例えば、「葛」の字の下の部分の形が変更されましたし、「辻」の字の「しんにょう」の点の数が一つから二つに増えました。実は、この変更は、戦後すぐ略字体として作られた文字を「本来の文字」とされている、中国の清の時代に編纂された「康熙字典」という漢字字典の字体に戻したものです。

Microsoft Windowsでは、Windows Vistaからこの変更が適用されています。ですので、Windows XPでは「辻」の点の一つ、Windows Vistaでは「辻」の点は二つになっています。なお、Windows XPでも、JIS2004に対応したフォントが配布されています。どの文字が変更されたかは、詳しくは、「JIS漢字コード表の改正について」^{*1}にあるPDFファイルをご覧ください。比較的によく使われる文字も多いので、自分の名前や住所で使っている漢字が入っているかもしれません。ただし、この変更はあくまでも、コンピュータ上で使う文字集合の字体の変更なので、通常の印刷物や手書きの文字などで前の字体の文字を使ってはいけないというわけではありません。

文字コードとは少し離れますが、ここで改行コードのお話もしておきます。

ASCIIコードの中には、改行を意味する制御文字があります。歴史的な事情により、0x0A (LF)と0x0D (CR)の二種類存在します。実は、OSにより、どのように改行を表現しているか異なってきます。少しややこしいのですが、WindowsはCRとLFの組合せで、Mac OSはCRで、UNIXやLinuxはLFで、それぞれ改行を表しています。

^{*1} <http://www.meti.go.jp/kohosys/press/0004964/> (閲覧日: 2009年12月22日)

2.3 Unicode

日本語だけでしたら、これで問題は解決するのですが、もちろん世界中には色々な言語が存在します。日本語以外にも文字がたくさん存在する言語があります。今までは、言語毎に文字コードを定めて使っていました。ですので、一つのファイルの中で複数の言語を混在させるのは非常に困難でした。しかし、1990年代初頭に Unicode と呼ばれるものが登場しました。これは、世界の全ての文字を共通の文字コードで表現するためのものです。

当初の Unicode は 2 バイトの中に全ての文字を収めようとしていました。日本語だけなら 2 バイトあれば十分そうですが、世界中の言語の文字を収めるためには、2 バイトだけではとても足りません。ですので、特に文字数が多い漢字の領域をできるだけ縮小させる作業が行われました。これを「ハンユニフィケーション」(Han Unification) と呼びます。具体的には、中国語、日本語、韓国語、ベトナム語で使われている (使われていた) 漢字の中で、字源が同じ漢字を一つにまとめてしまったのです。ご存知の方も多いと思いますが、日本語の漢字と中華人民共和国の統治下で使われている中国語の漢字 (簡体字) は、同じ意味、同じ由来の漢字でも字体が異なることが非常に多いです。また、台湾や香港、マカオなどで使われている漢字 (繁体字) や韓国語の漢字も字体が異なることが多いです。さらに、ベトナムで 20 世紀前半まで使われていた漢字もあります。(ベトナムは中国の影響で 20 世紀前半まで漢字が使われていましたが、現在は、ベトナム語話者でも、専門家くらいしかその漢字を理解できないようです。) このハンユニフィケーションの結果、二万文字ほどにまとめられた CJKV 統合漢字 (Chinese, Japanese, Korean, Vietnamese の頭文字を取っています。) というものができたのですが、これでは、各言語の文化を無視しているだけではなく、本来の目的である多言語の混在ができないという問題が生じてしまいました。また、全世界のコンピュータで使う以上、正確な規格を制定しなければならないのに、統合に関していくつも矛盾が存在すると言われていています。(詳しくは、「この言葉、あの言葉:ハンユニフィケーション (Han unification)」^{*2}や「Unicode の矛盾」^{*3}などをご覧ください。) 現在、この問題は、Unicode を 2 バイト以上に拡張することにより解決がはかられています。

この Unicode は、現在、OS の内部などでよく使われるようになってきていて、Windows でも、Mac OS X でも内部的には、Unicode が使われています。なお、実際の文字コード

^{*2} <http://blog.livedoor.jp/nrudt/archives/50979463.html> (閲覧日: 2009 年 12 月 22 日)

^{*3} <http://kanji.zinbun.kyoto-u.ac.jp/~yasuoka/kanjibukuro/unicode.html> (閲覧日: 2009 年 12 月 22 日)

としては、Unicode の文字コードを変換した UTF-8 というものがよく使われます。最近
は、日本語の表記でもよく UTF-8 が使われています。

- 「Unicode Consortium」*4

3 テキストエディタの種類

テキストエディタには、一般によく使われるものだけでも、いくつも存在します。この
節では、テキストエディタをいくつか紹介します。

3.1 メモ帳

Windows に標準でインストールされているテキストエディタです。コンピュータに
Windows がインストールされていれば必ずインストールされていることとなります。た
だ、メモ帳は、文字コードとして、Shift-JIS と UTF-8 しか対応していませんし、テキス
トエディタとして最低限の機能しか持っていません。

3.2 TeraPad

Windows でよく使われているフリーソフトのテキストエディタです。メモ帳が持つて
いない機能もきちんと持っています。通常の使用には十分です。この文書の執筆時点で最
新版は Ver.0.93 です。

- 「TeraPad」*5

3.3 Meadow

Unix でよく使われている GNU Emacs というテキストエディタを Windows に移植した
ものです。独特の操作体系を持っていて、慣れていないと使いにくいかもしれません。た
だ、拡張性に優れていて多言語処理も得意なので、その名前を覚えておいても損はないで
しょう。

- 「Meadow」*6

*4 <http://www.unicode.org> (閲覧日: 2010 年 2 月 8 日)

*5 <http://www5f.biglobe.ne.jp/~t-susumu/library/tpad.html> (閲覧日: 2009 年 12 月 22 日)

*6 <http://www.meadowy.org/meadow/> (閲覧日: 2009 年 12 月 22 日)

4 テキストエディタの使い方

この節では、3.2 節で紹介した TeraPad を元にして、テキストエディタの使い方を説明します。

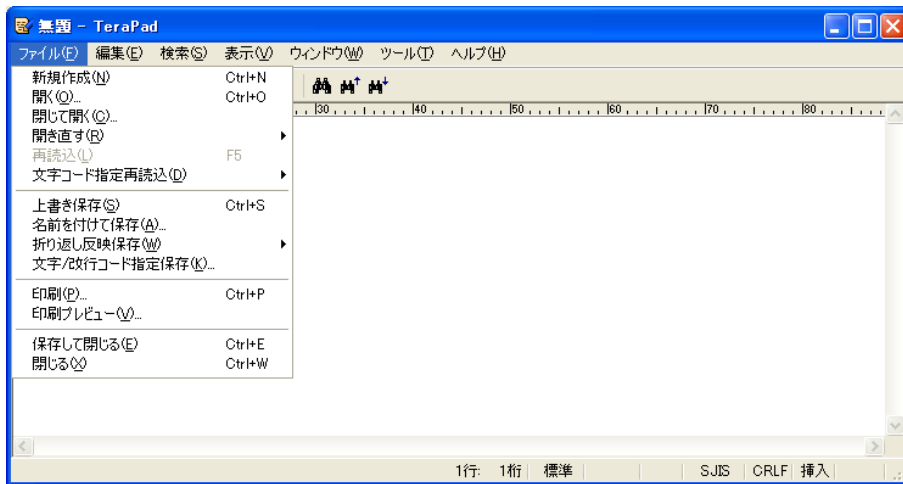


図 1 TeraPad の起動

図 1 は、実際に TeraPad を起動したところです。メニューを見ると分かりますが、ファイルを開いたり、保存したりする機能や編集などの機能は、通常のワープロソフトと同様なので、以下、テキストエディタ特有の使い方を中心に説明していきます。その他、TeraPad の詳しい使い方は、「TeraPad ヘルプ」^{*7}をご覧ください。

なお、TeraPad は標準で、長い一行はそのまま右に長く表示されます。これを避けるためには、図 2 ^{*8}のように、「表示」-「80 桁で折り返す」を選択します。

4.1 文字コードの認識

前述の通り、コンピュータ上では、文字は全て文字コードで記録されています。日本語で使われる文字、つまり、ひらがな、カタカナ、漢字などを表現できる文字コードとして、よく使われるものに以下のものがあります。

- ISO-2022-JP
- Shift_JIS

^{*7} <http://harpy.jp/delfy/tpadhelp/> (閲覧日: 2009 年 12 月 22 日)

^{*8} ここで開いているのは夏目漱石の「吾輩は猫である」です。 <http://www.aozora.gr.jp/cards/000148/card789.html> (閲覧日: 2009 年 12 月 22 日) から取得しました。

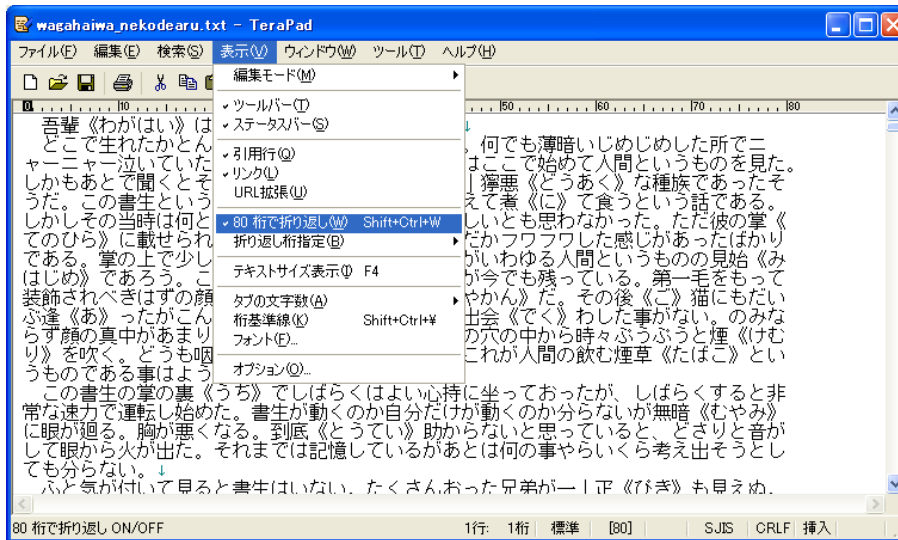


図 2 80 桁で折り返し

- EUC-JP
- UTF-8

Windows 標準のメモ帳は、Shift_JIS と UTF-8 にしか対応していません。ですので、ISO-2022-JP や EUC-JP で書かれた日本語のテキストファイルをメモ帳で開いてみても正常に表示されません。それに対して、TeraPad はこの四つの文字コード全てに対応しています。どの文字コードで書かれていても、TeraPad で開けば正常に表示することができます。ただ、テキストファイル自体に「このファイルは ISO-2022-JP で書かれています。」とか「このファイルは UTF-8 で書かれています。」とか書かれているわけではありません。どの文字コードが使われているかは、TeraPad 自体が、「どの範囲の数値が使われているか」「どういう数値の連続があるか」などの情報を元に判断しています。これである程度は認識できるのですが、短いファイルなどでは、誤認識されてしまうこともありますし、場合によっては、敢えて本来とは異なった文字コードで開いてみることもあるかもしれません。そのような場合は、TeraPad のメニューの「ファイル」-「文字コード指定再読込」から変更できます。また、「ファイル」-「漢字/改行コード指定保存」から、文字コードや改行コードを指定して保存することができます。

4.2 検索と置換

次に TeraPad の検索と置換の機能について説明します。

まず、検索ですが、メニューの「検索 (S)」-「検索 (F)」と選ぶか、ツールバーの検索

のアイコンをクリックすると図3のように「検索」のウィンドウが出てきます。検索したい文字列を入力してから「先頭から検索(O)」、もしくは、「次を検索(N)」をクリックすると検索した文字列が反転されて表示されます。「先頭から検索(O)」をクリックするとファイルの先頭から、「次を検索(N)」をクリックすると現在カーソルがある場所から、検索します。一度検索すると、「検索」のウィンドウは閉じられますが、メニューの「検索(S)」の「上を検索」や「下を検索」をすると検索を続行できます。

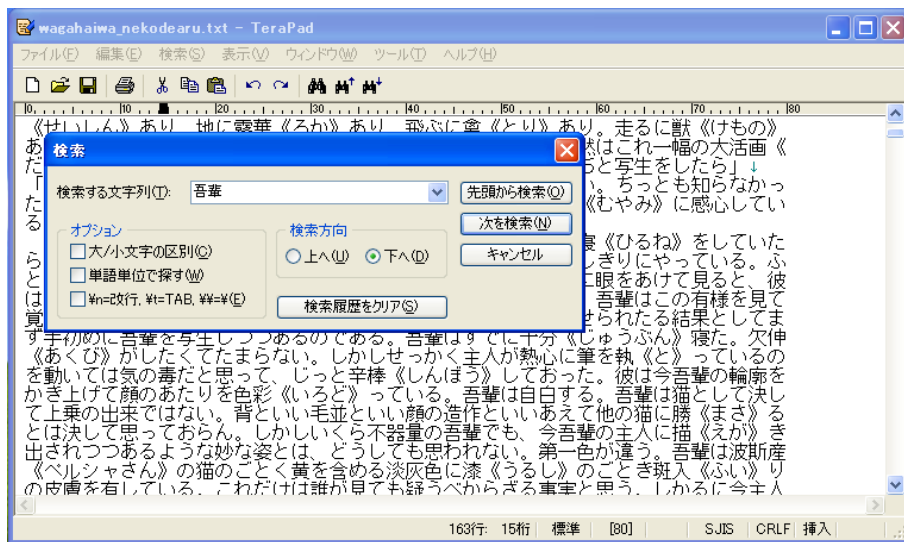


図3 検索

この検索のウィンドウには、以下のオプションがあります。それぞれチェックを入れると有効になります。

- 大/小文字の区別
- 単語単位で探す
- ¥n=改行、¥t=TAB、¥¥=¥

まず、「大/小文字の区別」ですが、アルファベットの文字列を検索する時、大文字小文字の区別はされません。つまり、「this」で検索すると、「This」や「THIS」などもマッチします。しかし、ここにチェックを入れると、大文字小文字が区別されます。「this」で検索すると「this」しかマッチしなくなります。これは、もちろん、日本語の文字だけで検索している時は無関係です。

次に、「単語単位で探す」ですが、これは、単語を認識して検索するものです。例えば、通常、「like」で検索すると「likely」や「likeness」「alike」などもマッチしますが、これに

チェックして検索すると、「like」という単語だけにマッチするようになります。単語の区切りは空白や改行、コンマ、ピリオドなどで判断しているようです。これは、日本語には機械的に判断できる単語の区切りのないで、日本語には対応していません。

さらに、三つ目の「¥n=改行、¥t=TAB、¥¥=¥」ですが、これは、改行やTABも検索できるようになるオプションです。例えば、「。¥n」として検索すれば、改行前の「。」(句点)だけにマッチします。なお、このオプションをチェックすると、¥記号に特別な意味が入るので、オプションを付けた状態で¥記号そのものを検索したい時は、「¥¥」のように二つ続けます。

置換を行う場合は、メニューの「検索(S)」-「置換(R)」を選びます。すると図4のように、「置換」のウィンドウが出てきます。

「検索する文字列」の部分は、上記の検索の場合と同様ですが、もう一つ、「置換後の文字列」を指定します。そして「先頭から検索(O)」「次を検索(N)」をクリックすると検索の場合と同様に、マッチした一つの文字列が置換されます。「連続置換(X)」や「すべて置換(A)」をクリックするとマッチしたすべての文字列が一度に置換されます。

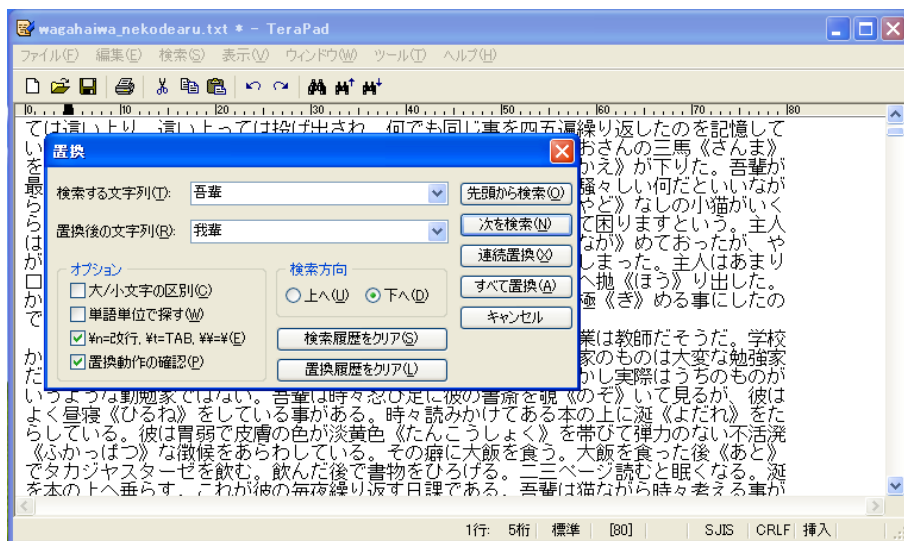


図4 置換

オプションには、以下の四つのものがあります。

- 大/小文字の区別
- 単語単位で探す
- ¥n=改行、¥t=TAB、¥¥=¥
- 置換動作の確認

最初の三つは、検索の場合と同様です。最後の「置換動作の確認」のチェックを入れると、「連続置換」や「すべて置換」の場合に、実際に置換するのか、確認しながら置換することができるようになります。

4.3 TeraPad からツールを使う

テキストファイルは、テキストエディタに限らず、様々なツールで柔軟に処理することができます。そのようなツールは、自分で手動で実行することもできますが、TeraPad の中から直接扱うこともできます。この節では、そのようなツールの使い方をいくつか紹介します。

4.3.1 フォルダを開く

まずは一例として、TeraPad から、現在編集中的ファイルが存在するフォルダをエクスプローラー (Explorer)^{*9}で開く方法を紹介합니다。

TeraPad のメニューから、「ツール (T)」-「ツールの設定 (T)」を選びます。すると、図 5 のようなウィンドウが出てきます。図 5 では、すでにいくつか設定をした状態になっていますが、まだ設定をしていない状態では、空欄になっています。新規に設定するためには、ここから「追加 (I)」をクリックします。すでに設定したツールを設定し直す場合には、「編集 (E)」をクリックします。

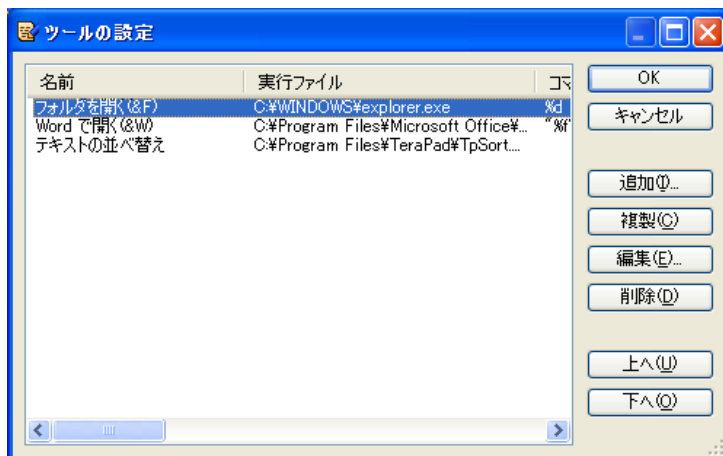


図 5 ツールの設定

^{*9} Windows でよく使われる Web ブラウザである Internet Explorer と Windows 標準のファイルやフォルダの操作ツールである Explorer は別のもので注意して下さい。この文書で「エクスプローラー」というと後者のものを指します。

図 6 が実際の編集画面です。それぞれの項目を順番に説明していきます。

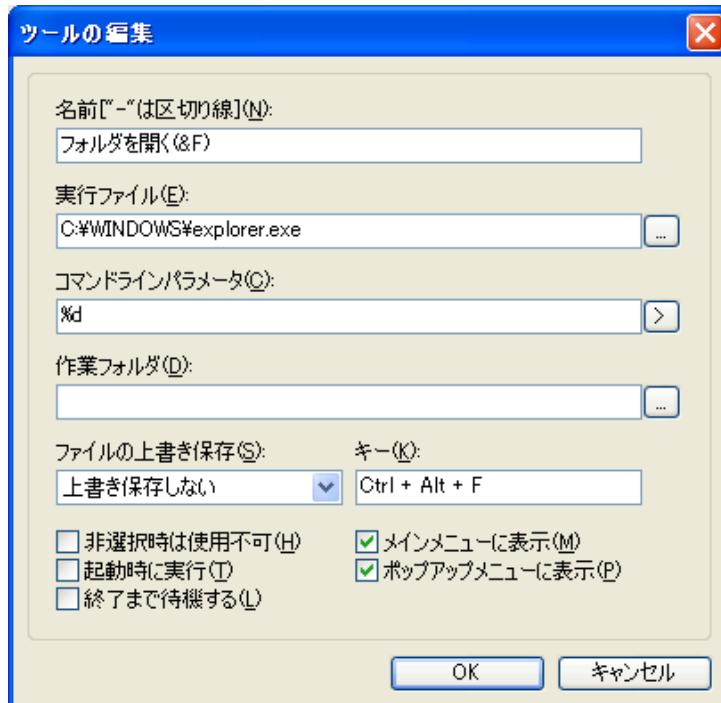


図 6 ツールの編集

名前 (N) ツールの名前を指定します。図 6 では、「フォルダを開く」としてあります。下の「メインメニューに表示 (M)」や「ポップアップメニューに表示 (P)」にチェックを入れていると、前者の場合は「ツール (T)」のメニューの中に、後者の場合は編集画面を右クリックした時のポップアップメニューの「ツール (S)」の中に表示されます。

また、この名前の中で、半角英字の前に「&」（アンパサンド）があると、メニューの中でその文字で項目を選択できるようになります。つまり、この例だと、「Alt + T」で「ツール」メニューを出した後、「F」を押すと「フォルダを開く」メニューを実行できます。

実行ファイル (E) メニューを選択した時に実行されるコマンドを指定します。この例の場合、エクスプローラーの本体は C:\Windows\explorer.exe にあるので、そのように指定しています。

コマンドラインパラメータ (C) コマンドに渡すコマンドラインパラメータ^{*10}を指定でき

^{*10} コマンドラインパラメータは、コマンドライン引数とも呼ばれます。

ます。

explorer.exe の場合、コマンドラインパラメータでフォルダを指定すると、初期状態で開くフォルダを指定することができます。この機能を使うと現在 TeraPad で開いているファイルが存在するフォルダをエクスプローラーで開くということが可能になります。そのためには「コマンドラインパラメータ」の欄に「%d」と入力します。この欄には、「%」が前に付いた英字をいくつか使うことができます。このような文字列を「特殊文字」と呼びますが、「%d」は、現在編集集中のファイルが存在するフォルダに置き換わります。例えば、現在編集集中のファイルが c:¥Temp にあったとすると、%d は“c:¥Temp”に置き換わり、それが explorer.exe にコマンドラインパラメータとして送られることとなります。他の「%が前に付いた英字」にどのようなものがあるかは後述します。

ファイルの上書き保存 (S) ツールを実行する際に現在編集集中のファイルを保存するかどうか選択します。「上書き保存しない」「確認メッセージを表示する」「上書き保存する」の三つから選択できます。

キー (K) ツールを起動するためのショートカットキーを指定できます。この例の場合には、Ctrl キーと Alt キーを押しながら F キーを押すとツールが起動します。他のメニューと重複しているかどうかはチェックされていないようなので、注意が必要です。

なお、ここで説明していない項目もありますが、詳細は「TeraPad のヘルプのページ」^{*11}をご覧ください。

また、コマンドラインパラメータで使える特殊文字には以下のようなものがあります。

- %f 編集集中のファイルのフルパス^{*12}
- %n 編集集中のファイルの拡張子付きファイル名
- %b 編集集中のファイルの拡張子なしファイル名
- %d 編集集中のファイルが存在するフォルダ名
- %s 現在選択している文字列
- %t 対象範囲を書き出した一時ファイルのフルパス

^{*11} <http://harpy.jp/delfy/tpadhelp/tool.html> (閲覧日: 2010 年 1 月 31 日)

^{*12} ドライブ名、フォルダ名、ファイル名をすべてつなげたもの。

4.3.2 編集中のファイルを他のアプリケーションで開く

この特殊文字の機能を使うと、現在編集中のファイルを MS Word や MS Excel などの他のファイルで開くということもできます。

例えば、以下のように設定します。

名前 Word で開く

実行ファイル C:\Program Files\Microsoft Office\Office12\WINWORD.EXE

コマンドラインパラメータ "%f"

このようにすると、現在編集中のテキストファイルを Word で開いて編集することができます。実行ファイルの部分には、Word の実行ファイルそのものを指定します。Word の実行ファイルは、WINWORD.EXE という名前になっています。上記の例は、MS Office 2007 のものですが、バージョンによって、場所が異なります。また、コマンドラインパラメータは「編集中のファイルのフルパス」を表す特殊文字 %f を使っています。これで、現在編集中のファイルのフルパスが WINWORD.EXE に渡されるのですが、フルパスに空白が含まれている場合に上手く渡されないことがあるので、「"」（二重引用符）でくくっています。

これで、現在編集中のファイルのテキストファイルを、引き続き Word で編集できるようになります。例えば、一通り TeraPad で文章を書いてから、後でまとめて Word でフォント装飾やスタイルを付けるというような使い方ができるようになります。

Excel でも、EXCEL.EXE という実行ファイルを指定することにより、同様のことができます。

4.3.3 ソートを行う

次に、ツールを使ってソートを行う方法を紹介します。

例えば、単語のリストがあって、それをアルファベット順に並べたい時や、数字のデータを大きさ順に並べたい時には、ソートの機能を使います。ソートを行うためのツールはいくつか存在しますが、ここでは、「TpSort」^{*13}を使ってみます。

使い方の詳細は上記の URL に記されていますが、「ツールの設定」で以下のように設定します。なお、ここでは、TpSort を TeraPad の標準のインストールフォルダ (C:\Program Files\TeraPad) にインストールしたと仮定しています。

*13 <http://harpy.jp/Tategoto/terapad/TpSort.html> (閲覧日: 2010 年 2 月 3 日)

名前 テキストの並べ替え

実行ファイル C:\Program Files\TeraPad\TpSort.exe

TpSort は、プログラム内部で TeraPad と編集内容の引き渡しをしているようで、編集中のファイル名などをコマンドラインパラメータで指定する必要はありません。ただ、ファイル名以外の選択項目があって、それを指定すると TpSort の標準の動作を変更することができます。詳細は上記の TpSort の Web ページのコマンドラインパラメータの項目をご覧ください。

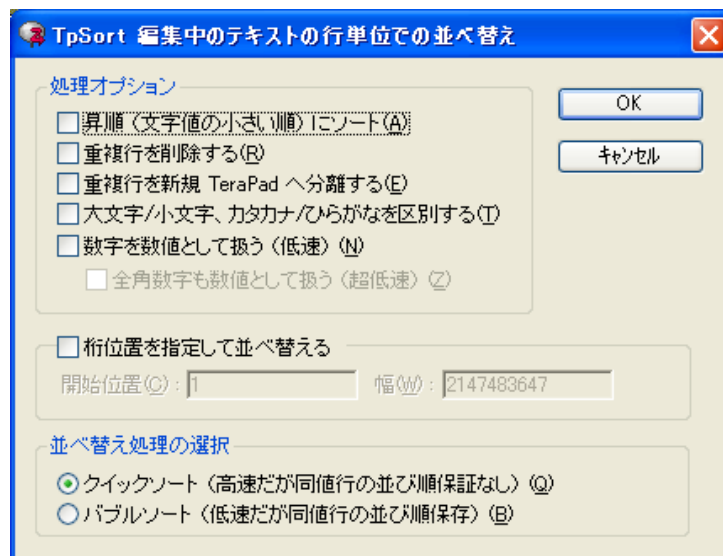


図 7 TpSort

図 7 は、ツールから「テキストの並べ替え」を実行して、TpSort のウィンドウが表示されたところです。ここで、処理オプションなどを選択して「OK」をクリックすると TeraPad で編集中のファイルに対してソートが実行されます。ソートの結果は TeraPad の編集画面に自動的に反映されます。

ソートを行うためには、そもそも項目をどのような順序に並べるのか、ということを決める必要があります。通常は、アルファベット順や数値順に並べることが多いと思います。ただ、アルファベットなどの文字は、コンピュータの中では文字コードで管理されているので、基本的には文字コード順に並べることになります。したがって、例えば、アルファベットなら、ASCII コードの数値順に並びます。日本語の文字を使っても、同様にその文字コードの順番に並びます。

ただし、「通常の辞書のように大文字小文字を区別しないで並べたい」というような希

望もあると思います。そのために利用できる処理オプションがいくつかありますので、以下でそれを説明します。

まず、一番上の「昇順 (文字値の小さい順) にソート」です。これにチェックを入れると、「文字値」、つまり文字コードが小さい方から順番 (昇順) に並びます。チェックを入れないと大きい方から順番 (降順) に並びます。

次に四つ目の「大文字/小文字、カタカナ/ひらがなを区別する」についてです。これにチェックを入れると、並べ替えの際に、大文字/小文字やカタカナ/ひらがなが区別されるようになります。例えば、A, a, B, b の四つの項目がある場合、通常の辞書などでは、大文字/小文字を区別せずに、A, a, B, b と並べます。つまり、大文字と小文字を同一視し、同じ場所に置いていることになります。それに対して、この項目にチェックを入れて区別するようにすると、この四つの項目は、A, B, a, b と並べられます。つまり、大文字と小文字を同一視せず、そのまま文字コード順に並べることになります。

最後に、「数字を数値として扱う」について説明します。例えば、1, 2, 10, 11 という四つの項目がある場合、そのまま並べ替えると、1, 10, 11, 2 という順番に並びます。これは、文字コード順の並びではありますが、数値順ではありません。そこで、この項目にチェックを入れると、数値順に、1, 2, 10, 11 と並びます。実際に、数値の情報を並べる時に利用できますが、通常の並べ替えよりも処理に時間がかかるようになるようです。

4.3.4 検索と置換を行う

次に、文字列の検索と置換を行うツールを紹介します。元々、TeraPad には、検索と置換の機能が付いていますが、機能として貧弱です。ここでは、外部のツールを使って、テキストの検索と置換を行う方法を紹介します。

テキストの検索と置換を行うツールはいくつも存在しますが、ここでは、「Devas(ディーヴァス)」^{*14} というツールを使ってみます。

図 8 は、Devas を TeraPad のツールとして設定したものです。Devas の本体を C:\Program Files\Devas にインストールしています。

設定のポイントは以下の通りです。

名前 Devas で検索/置換

実行ファイル C:\Program Files\Devas\Devas.exe

コマンドラインパラメータ %t.txt

^{*14} <http://gimite.net/pukiwiki/index.php?Devas> (閲覧日: 2010 年 2 月 7 日)

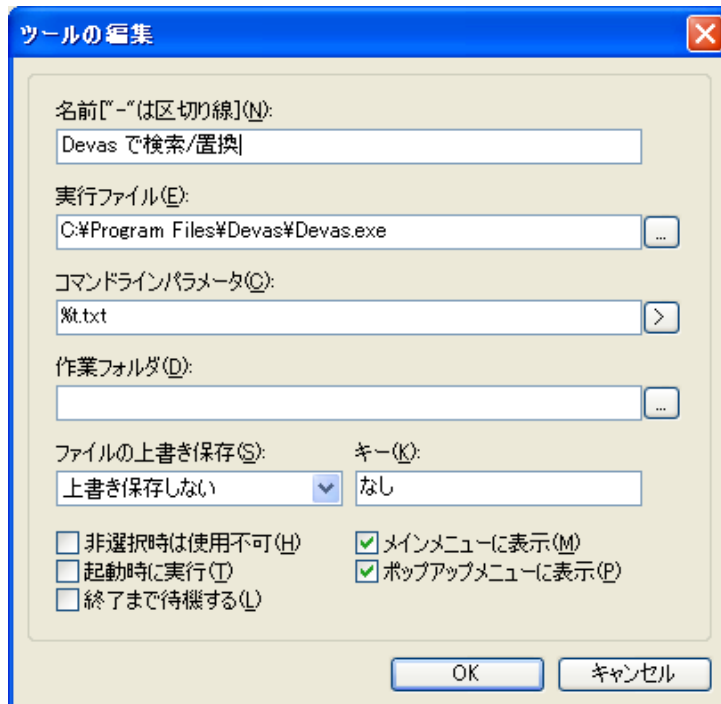


図 8 Devas の設定

ここで、コマンドラインパラメータの特殊文字“%t”について説明します。これは、「現在編集集中のファイル、もしくは、選択範囲を書き出した一時ファイルの名前」という意味を持っています。つまり、この特殊文字を使っていると、TeraPad の中で、選択中の範囲があればその範囲を、なければ現在編集集中のファイル全体を別のファイルにして自動的に書き出されます。%t は、そのファイルのファイル名に置き換わります。この例では、.txt という拡張子も付けています。これで、Devas.exe が、この一時ファイルをコマンドラインパラメータとして実行されることとなります。Devas.exe は、コマンドラインパラメータとして検索/置換を行うファイル名をとるので、この一時ファイルが検索と置換を行う対象になります。

これで、「Devas で検索/置換」を実行すると、Devas のウィンドウが出てきます。実際の検索/置換の対象である「対象ファイル/フォルダ (A)」の部分に、この一時ファイルのファイル名が入った状態になっています。さらに、「検索文字列 (S)」のところに入力した文字列を入れて、「検索 (F)」をクリックします。すると、別のウィンドウで検索結果が表示されます。

図 9 は実際に「吾輩は猫である」のテキストで「吾輩」という文字列を検索してみたところです。このように、前後の文字列も含めて、一行一項目で表示されます。さらに、検

検索結果のウィンドウの下の「置換文字列 (S)」のところに置換先の文字列を入れると置換を行うことができます。検索結果のウィンドウの左端のチェックボックスで、項目毎に置換をするかどうか決めることができます。

ただし、この置換は一時ファイルの方で行われます。それだと、現在編集のファイルには、関係ないように思えるかもしれませんが、Devas を終了すると、自動的に TeraPad の編集画面にその内容が反映されます。特殊文字 %t は、編集のファイルの内容を一時ファイルに書き出すだけでなく、外部アプリケーション終了後に、編集のファイルにその一時ファイルの内容を書き戻す機能も持っています。

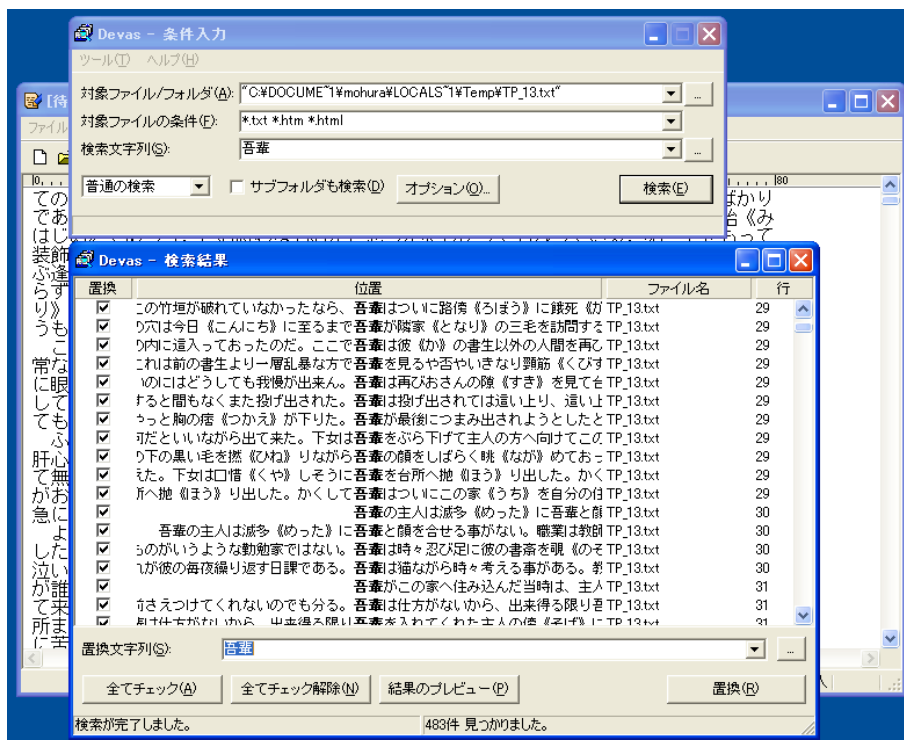


図 9 Devas で文字列の検索

4.3.5 どのようなツールが使えるのか

最後に TeraPad で利用できるツールにはどのようなものがあるのか、紹介します。

まず、TeraPad 用に作成されたツールが色々と存在しています。それをまとめたページをいくつか紹介します。

- 「TeraPad Tools」^{*15}
- 「Tategoto is here. - With TeraPad」^{*16}
- 「Vector：ダウンロード Windows > 文書作成 > テキストエディタ > TeraPad」^{*17}

また、この文書で紹介したエクスプローラーや MS Word のように、コマンドラインパラメータとして、ファイル名やフォルダ名を取るようなアプリケーションの場合は、現在編集中のファイルのファイル名などを渡すことにより、便利に使うことができます。

さらに、Devas の例のように、コマンドラインパラメータとしてテキストファイルを取り、それに対して何らかの処理を行い、その処理結果を元のファイルに保存するようなアプリケーションの場合は、特殊文字 “%t” を用いることにより、ツールとして使うことができます。

ただ、これに付いてはここのアプリケーションのマニュアルなどを見る必要がありますが、コマンドラインパラメータを取らないようなアプリケーションもあります。例えば、処理するファイルを選択するためには、コマンドラインパラメータではなく、メニューから選択しなければならないような場合は、TeraPad のメニューから起動させることはできませんが、ツールとして便利に使うことはできません。

^{*15} <http://www5f.biglobe.ne.jp/~t-susumu/library/tptools.html> (閲覧日: 2010 年 2 月 7 日)

^{*16} <http://harpy.jp/Tategoto/terapad/> (閲覧日: 2010 年 2 月 7 日)

^{*17} <http://www.vector.co.jp/vpack/filearea/win/writing/edit/terapad/> (閲覧日: 2010 年 2 月 7 日)