

電卓の最終形!!

単位変換機能付電卓

「UNITILO」は単なる「電卓を真似た電卓ユーティリティ」とは一線を画すモノです。その画期的な機能とは...

例えば、円の面積(半径の2乗×円周率3.14)、あるいは、時速×時間で距離を求める際の計算式に単位を付加すると、

```
5mm^2*3.14 ..... 78.5 [mm2]
112[km/h]*1:42:00 ..... 190.4 [km]
```

なんと、単位付で計算結果が出ます。勿論、体積、速度、力学、電気、放射能関係の単位もサポート。だから、3カラットのダイヤモンドを初速度2ノット、3分51秒後の速度4km/hで進める力の、その力の方向に500オングストローム動かすときの仕事を6分間で行う割合と同等の消費電力の4アットアンペアの直流電流を流したときの電圧(め、滅茶苦茶な例...)、

```
3ct*((4[km/h]-2kt)/0:3:51)*500[Å]/6min/4[aA]..... 7.4`15424082090748757` [V]
```

なんて計算もこなしてしまう。まさに、電卓を超えた電卓!?

他にも、計算値は全て文字列で処理しているので、10桁?、100桁?、いえいえ、(CPUパワーとメモリが許せば)ほぼ無制限の桁数を扱えます。また、除算結果が循環小数で出てきますので、「1÷3」が「0.33333」ではなく、「0.3」の形で結果が出ますし、更には循環小数での数値入力、仮分数、帯分数での計算結果出力、更に更にn進数での入出力と、こんな電卓、今までありました?(あったりして...)

尚、「UNITILO」は、「うにちろ」とでも呼んで下さい。

インストール、使用方法

UNITIL15.ZIP を解凍して下さい。

解凍したファイルの中から「Unitilo.exe」をデスクトップにでもコピーして使ってください。インストールプログラム也没有なので。

起動はそのままクリックするだけ。

レジストリも汚しません。INIファイルも作りません。よって、アンインストールは「Unitilo.exe」を削除するだけです。

簡単な計算

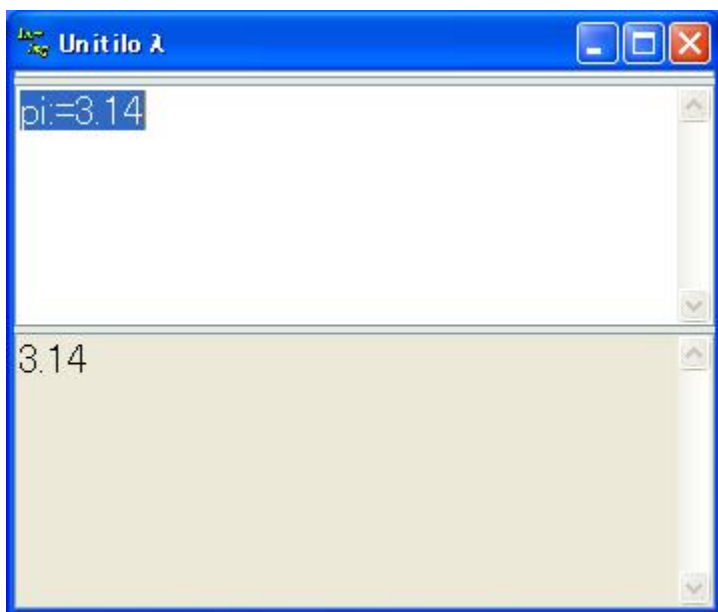
まず、「1+1」の簡単な計算を行います。ウィンドウが上下2つに分かれていますが、上欄に計算式を入力してリターンキーを押すと、下欄に計算結果が表示されます。

下記の例では、「1+1」を入力し、リターンキーを押しています。



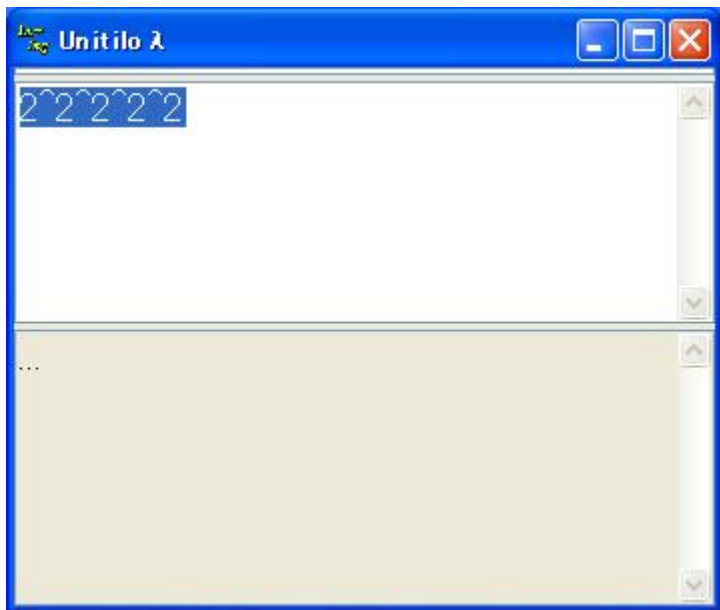
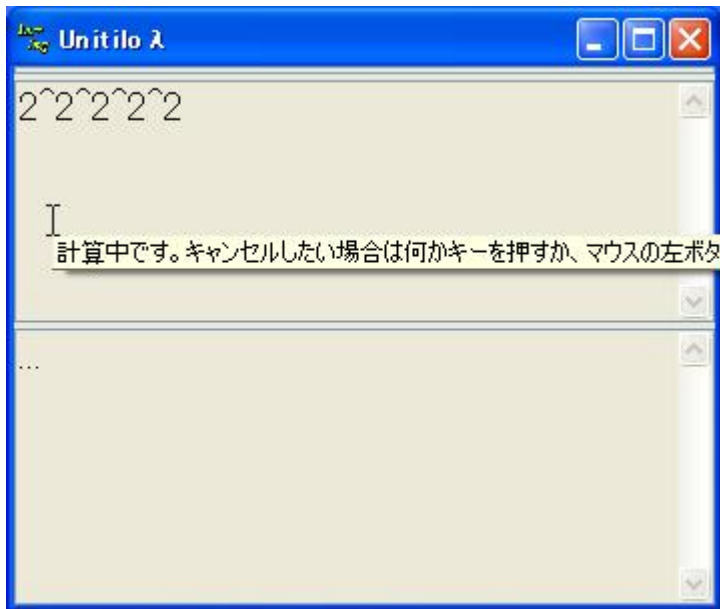
変数へ値の代入

計算式には変数を使用できます。以下の例では変数piに3.14を代入しています。変数は「UNITILO」を終了するか、「!reset」コマンドを実行するか、他の値を代入するまでは記憶されます。



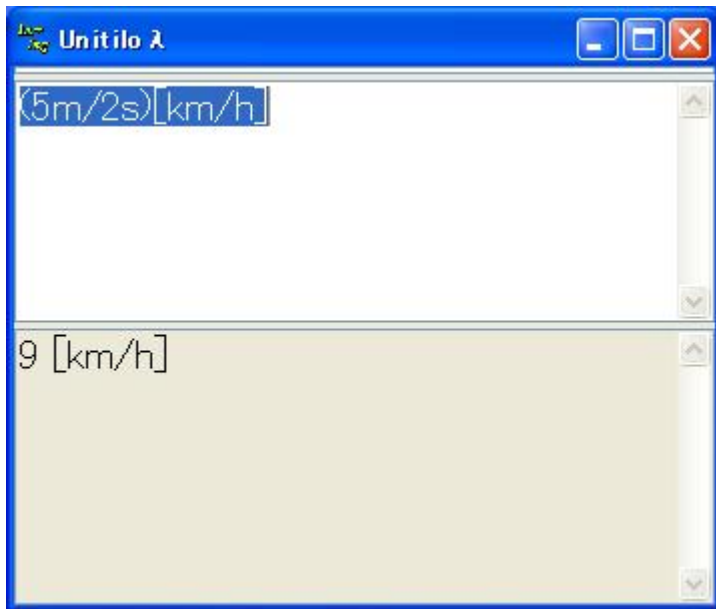
計算の中断

巨大な値や無理数などを計算させた場合、UNITILOは計算を永遠と続けますが、その間、途中経過を出力する場合もあれば「...」のみで結果を出さない時もあります。その場合、何かキーを押すか、上欄をマウスでクリックすれば、UNITILOは計算を中断します。なお、計算中は上欄背景がボタン色(多分、灰色)になり、計算が完了するとウィンドウ背景色(多分、白色)に変わります。



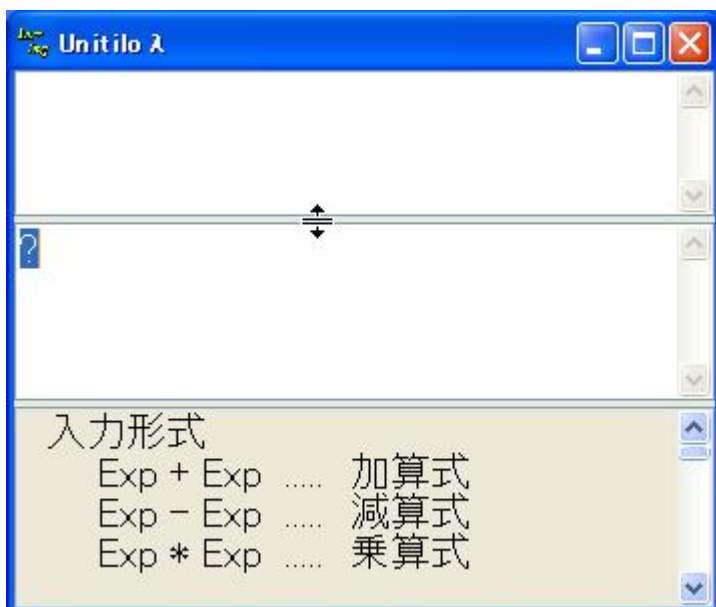
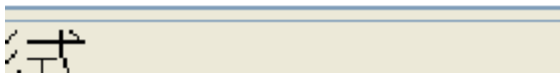
単位を使った計算

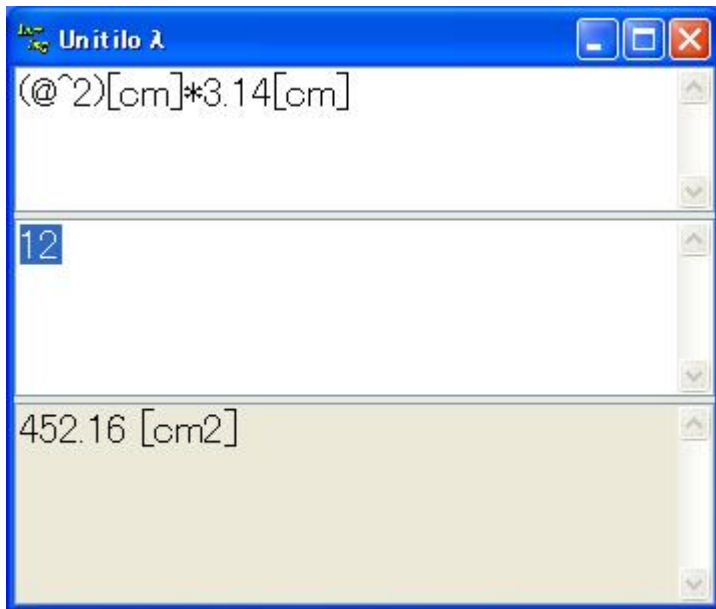
計算式には単位を付加することができます。扱える単位は次項の[仕様](#)を参照、または上欄に「?」でリターンキーを押すと下欄に扱える単位を表示します。ここでは距離と時間(秒)を元に秒速を計算し、これを時速に変換する例を示します。単位は数値の後に付加します。ここでの計算式「5m÷2s」を計算させると結果が秒速「m/s」で出てしまいますので、これを時速に変換するのが括弧の後の単位「(...) [km/h]」です。単位の指定は基本的には「[]」でくくって指定しますが、「/」や「*」を含んでいない単位の場合は、省略してもかまいません。



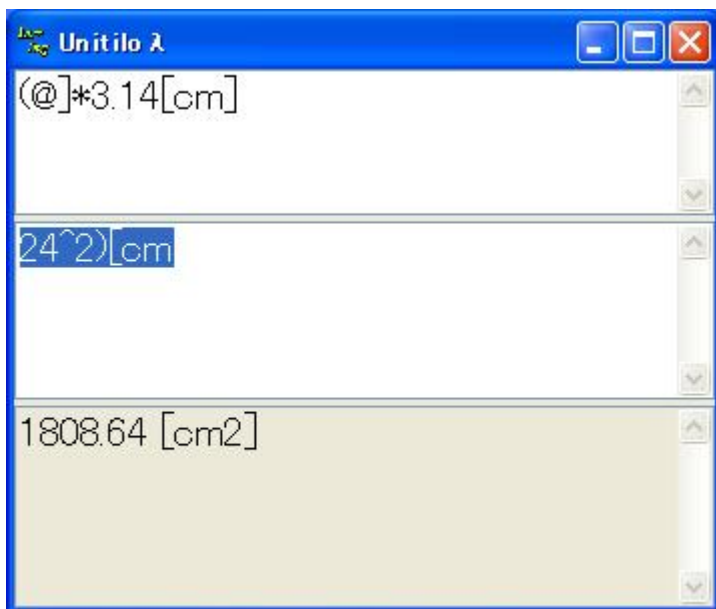
λ(ラムダ)式

λ式(と威張るほど、複雑な置換は出来ませんが...)を使った計算をする場合、タイトルバー下のバーをクリックして上欄に「@」文字を含む計算式、中欄に「@」の置換式を入力して使用してください。





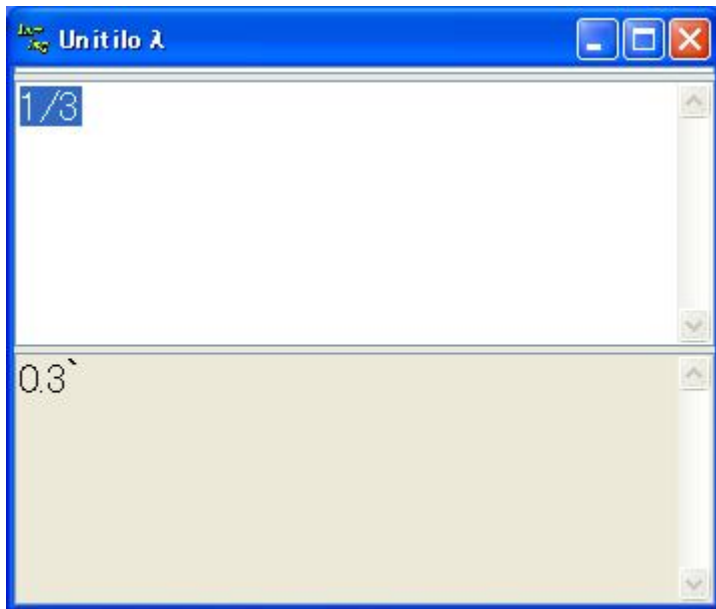
注意すべき点は、あくまで、「@」を中欄の計算式に置換してから計算するのであって、中欄式の計算結果が置き換わる訳ではありません。以下にその極端な例を示します。



置換式を使う際は副作用を考慮して、「(@)」と括弧で括って使用するのも良いです。

計算結果の分数出力

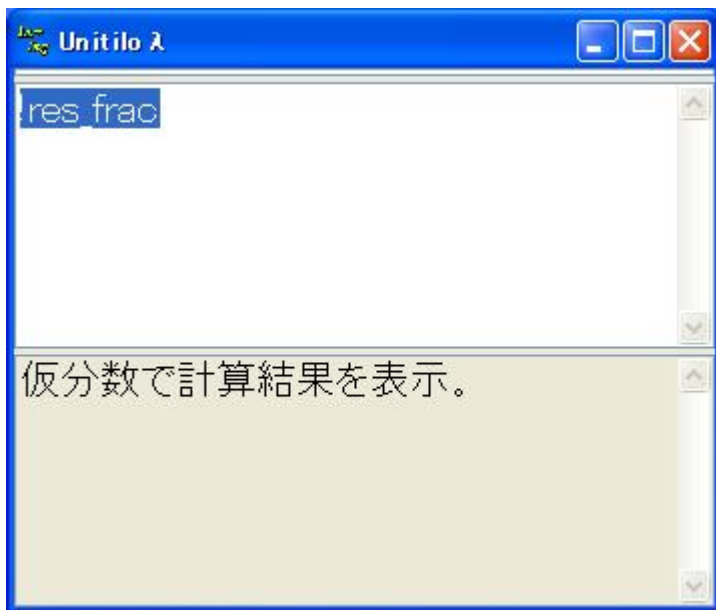
最初にも説明しましたが、UNITILOでは割り切れない小数点を循環節で出力します。



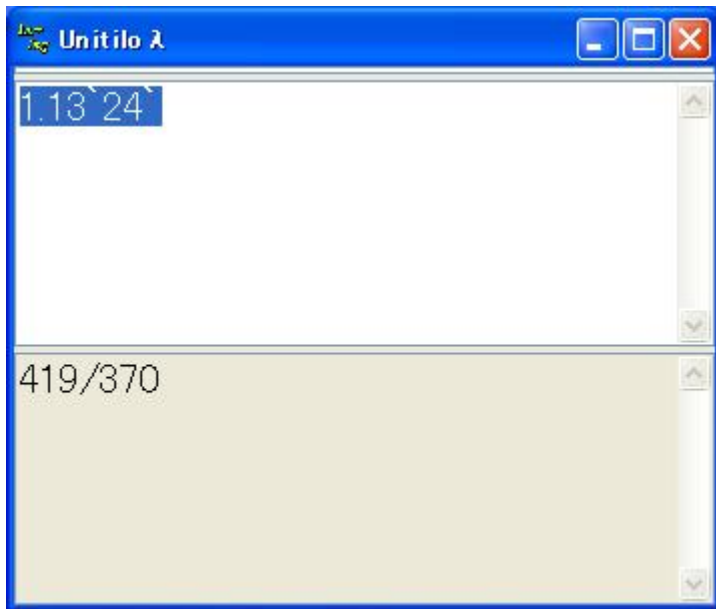
「 $1 \div 3$ 」の計算結果「 $0.333\dots$ 」或いは「 $0.\dot{3}$ 」をバッククォート「```」を付加した形で表示します。逆にバッククォート「```」で循環節数値を入力する事も出来ます。

その前に、計算結果を分数で出力する設定をしましょう。上欄にコマンド「`!res_frac`」リターンキーで以後、計算結果に小数が含まれる場合、分数で結果を出力します。

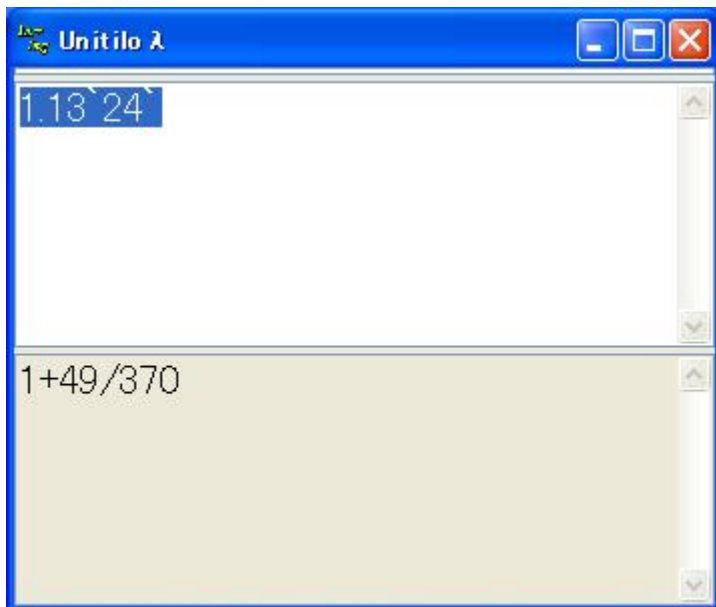
なお、帯分数は「`!res_mixfrac`」、元の小数出力は「`!res_dec`」コマンドを使います。



ちょっと複雑に「 $1.\dot{1}324$ 」を入力します。

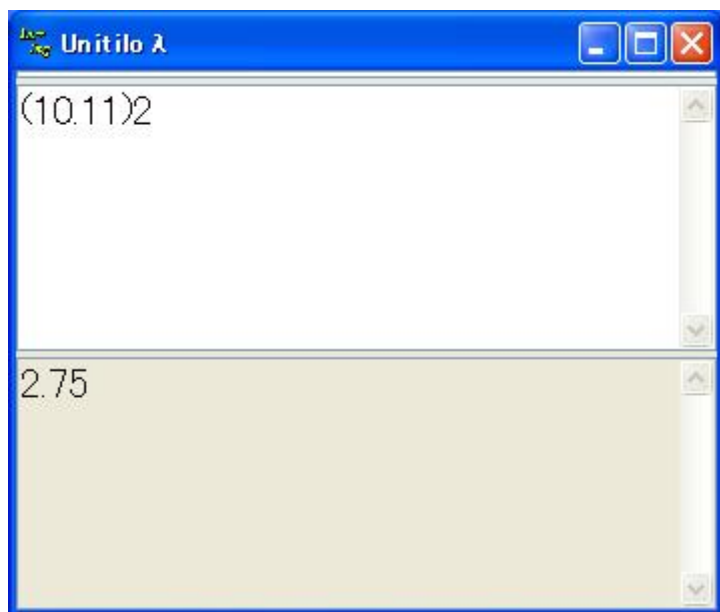


結果が仮分数で表示されます。帯分数「!res_mixfrac」に変更して同じ値を入力してみます。なお、「↑」キーで前の入力値を選択(履歴入力)する事も出来ます。



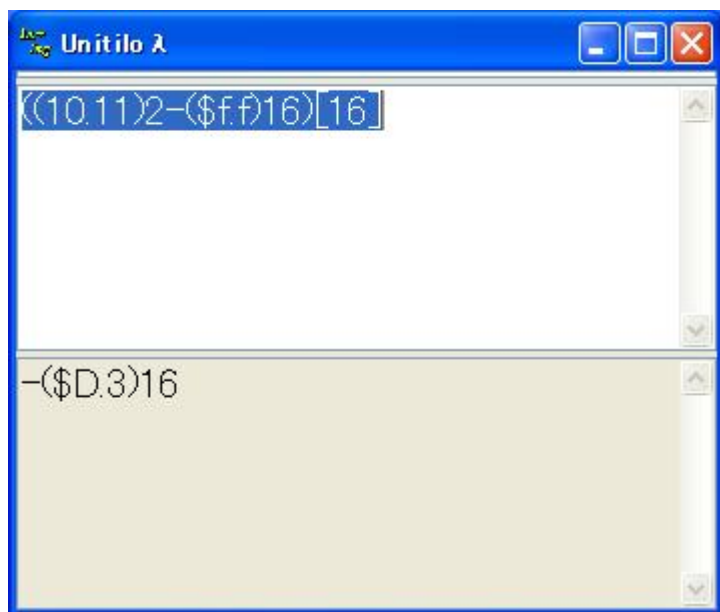
n進数の入力、結果表示

n進数(nは2から最大36)を入力する場合、括弧「()」の後に基数を指定する形で入力します。以下は二進数の入力例です。



括弧内は数値であり、式は入力できません。マイナスの場合も負数「-」は括弧の外に指定してください。11進数以上は英文字a～zまたはA～Zで数値を表しますが、変数と区別するために接頭子「\$」を付加します。なお、内部の演算は10進数で行う為、n進数指定はフロントエンドと考えてください。n進数の出力は単位形式「[]」でくくっての指定になります。

以下に演算結果を十六二進数で出力する例を示します。



終了

UNITILOを終了させる場合は、ウィンドウ右上の「x」ボタンをクリックしてください。

仕様

データ型

「UNITILO」内の数値、および変数値は文字列で処理しています(内部的には符号付の分数型文字列で

処理しています)。よって演算する際の数値桁はDelphiのString型の最大長(2^{30} 文字)を扱えます。とはいものの、正直言って、そんな長い値は試したことがありません。式の入力や結果表示に使用しているテキストボックスコントロールがどこまで耐えるのか不明ですし、それだけ長い数値ですと、演算処理も(というより、演算以前の文字列データの受け渡しすらも)めちゃくちゃ遅くなりますので、とりあえず、100桁を限度と思って使ってください。また、累乗の計算処理に関しては、一部Integer型を使用しているので、指数値に巨大な値を入力するとオーバーフローエラーで落ちる・・・と思います。

数値の入力フォーマットは、整数、小数(循環節の場合はバッククォート「```」を付加)、n進数(a～zまたはA～Zの英文字表記)が利用できます。但し、n進数は10進数に変換して計算されます。また、他に例外として「hh:mm:ss」の時間形式での入力出来ます。ただし、「hh:mm:ss」時間形式は秒単位として処理されます。変数名は英文字で始まる英数文字列で、英大文字、小文字は区別されます。変数には数値の他に単位も付加されます。

演算子他

「UNITILO」で扱える演算子は、加算「+」、減算「-」、乗算「*」、除算「/」、累乗「^」の二項演算子、マイナス値「-」の単項演算子、変数へ値を代入する演算子「:=」、式の評価順を決める括弧「()」および、単位を表す演算子?のみです。三角関数、超越数は未対応です。

式の評価順序は一般的なものと同じで、「括弧」、「累乗」、「乗算、除算」、「加算、減算」の順で優先順位が高く、左から順に評価します。ただし、累乗だけは右から順に評価されるため、「 $2^2^2^2$ 」を計算させた場合、「 $((2^2)^2)^2$ 」ではなく、「 $(2^{(2^{(2^2)})})$ 」の順で評価されますので、結果も「256」ではなく「65536」となります。

単位

「UNITILO」で扱える単位は以下の通りです。また、単位にはSI接頭語も付加できます。

記号	計算単位
[m]	長さ
[s]	時間(秒)
[min]	時間(分)
[h]	時間(時)
[g]	質量
[t]	質量
[K]	温度
[°C]	温度(セルシウス度)
[mol]	物質質量
[A]	電流
[cd]	光度
[Gal]	加速度(重力加速度)
[N]	力
[J]	仕事、熱量
[ct]	質量(宝石)

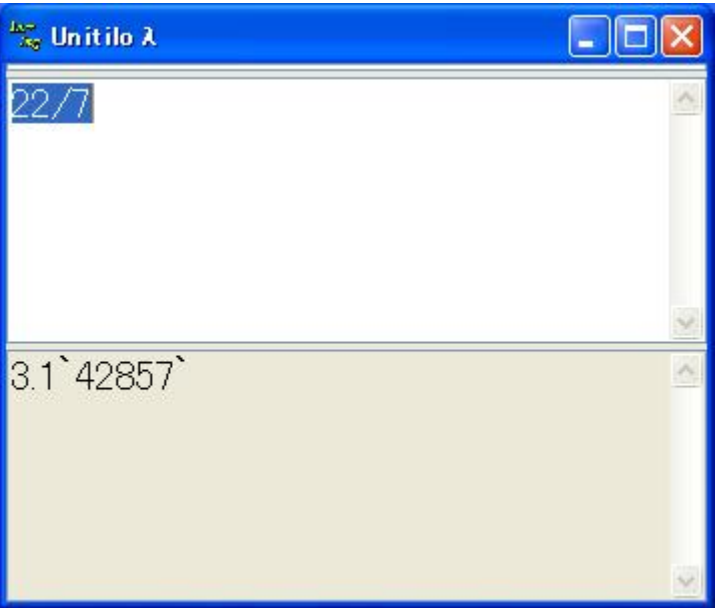
[mon]	質量(真珠)
[oz]	質量(金貨)
[Pa]	圧力
[bar]	圧力(気圧、流体)
[atm]	圧力(気圧)
[Torr]	圧力(生体内圧)
[mHg]	圧力(血圧測定)
[l]	体積
[a]	面積(土地)
[M]	長さ(海面、空中)
[Å]	長さ(光学、結晶学)
[AU]	長さ(天文)
[pc]	長さ(天文)
[Hz]	周波数
[W]	仕事率、電力
[cal]	熱量(栄養関係)
[C]	電荷、電気量
[V]	電圧、起電力
[F]	静電容量
[Ω]	電気抵抗
[S]	コンダクタンス
[H]	インダクタンス
[Wb]	磁束
[T]	磁束密度
[Bq]	放射能
[Ci]	放射能
[R]	照射線量
[Gy]	吸収線量、カーマ
[rad]	吸収線量
[Sv]	線量当量
[rem]	線量当量
[P]	粘度
[St]	動粘度
[kt]	速度(航海、航空)
[km/h]	速度
[m2]	面積
[m3]	体積

[m/s2]	加速度
[s-1]	毎秒
[R/s]	照射線量率
[Gy/s]	吸収線量率

言い分け: SI接頭語については、単位に付加するのに参考となるものが無かったため、適当に処理しています。「Mm(メガメートル)」とか「dm(デシメートル)」とかいう単位は作れませんが、「zm(zeptメートル)」とか「ym(yoktoメートル)」は出来ちゃいます。

演算

データ型の項でも説明したとおり、数値データは分数で保持しており、分数のまま演算されます。そして計算式を全て演算後、最後に分数を展開します。分数が割り切れない場合、例えば「22/7」は循環小数で結果が出力されます。また、コマンド指定により仮分数、帯分数で結果を出力することも出来ます。



循環節は、数字の後のバッククォート「`」で表示され、上記の場合、「142857」が循環節となります。

また、前記「[演算子他](#)」の項には(意図的に)記入しなかったのですが、指数に分数を指定して、累乗根を計算することも出来ます。例えば「 $\sqrt{2}$ 」を計算したい場合、「 $2^{(1/2)}$ 」とすれば、計算できます。でも、意図的に記入しなかったのは「 $2^{(1/2)}$ 」を試していただければおわかりの通り、恥ずかしくなるくらい計算が遅いかつ永遠と演算を続ける為、例えば累乗根同士を加減算すると、ほぼ永遠に答えが出てきません。小数点の切り捨て処理の仕様に関してはUNITILOの今後の課題となっております。

コマンド

その他、補助機能として、以下のコマンドをサポートしています。計算式と同様、上欄に入力、リターンキーを押して実行してください。

コマンド	機能
!help ?	ヘルプを表示
!reset	変数値及び履歴をクリア
!version	UNITILOのバージョン表示

!res_frac	計算結果を仮分数で表示
!res_mixfrac	計算結果を帯分数で表示
!res_dec	計算結果を通常(小数)表示

隠し機能として「!sourcecode」コマンドでUNITILOのソースコード(Delphi形式等)が出力されます。あくまで隠し機能なので詳細説明は有りません。悪しからず。

計算式の履歴

入力計算式は最大20個が履歴として保持され、変数と同様、「UNITILO」を終了するか、「!reset」コマンドを実行するまで保持されます。

履歴は「↑」「↓」または「PageUP」「PageDown」キーで入力できます。

「UNITILO」の配布。利用に当たっての条件、連絡先等

本ソフトウェアはフリーソフトウェアであり、ソフトウェアを配布する際の制約は有りません。

本プログラム動作による損害については何の責任も負いません。

本プログラムの著作権は以下の者が有しています。実行ファイル「Unitilo.exe」のプロパティ - 詳細からも確認できます。

(C) 著作権者 小野寺和伯, 発行年 1998-2025年

連絡先 ono4@nifty.com

尚、電子メール[\[ono4@nifty.com\]](mailto:ono4@nifty.com)によるプログラム、操作方法に関しての質問、要望、不具合等に対する返信は基本的に、一切致しません。(返答をしないだけで、要望をいっさい聞き入れないという訳ではありません、多分...

バージョンアップ、不具合、その他最新情報

- Ver.1.5.0.0 循環小数入力。分数出力対応。
- Ver.1.4.0.0 基数変換処理対応。
- Ver.1.1.0.0 VisualBasicからDelphiへ移植。

ホームページ <http://ono4.a.la9.jp/> を覗けば、最新版が有るかも!

参考文献等

松山 裕,『やさしい計量単位の話』,財団法人省エネルギーセンター
堀場 芳数,『円周率 π の不思議』,『対数 e の不思議』,講談社 BLUE BACKS
近藤 嘉雪,『yaccによるCコンパイラプログラミング』,ソフトバンク
奥村 晴彦,『C言語によるアルゴリズム事典』,技術評論社

[文書の終わり]