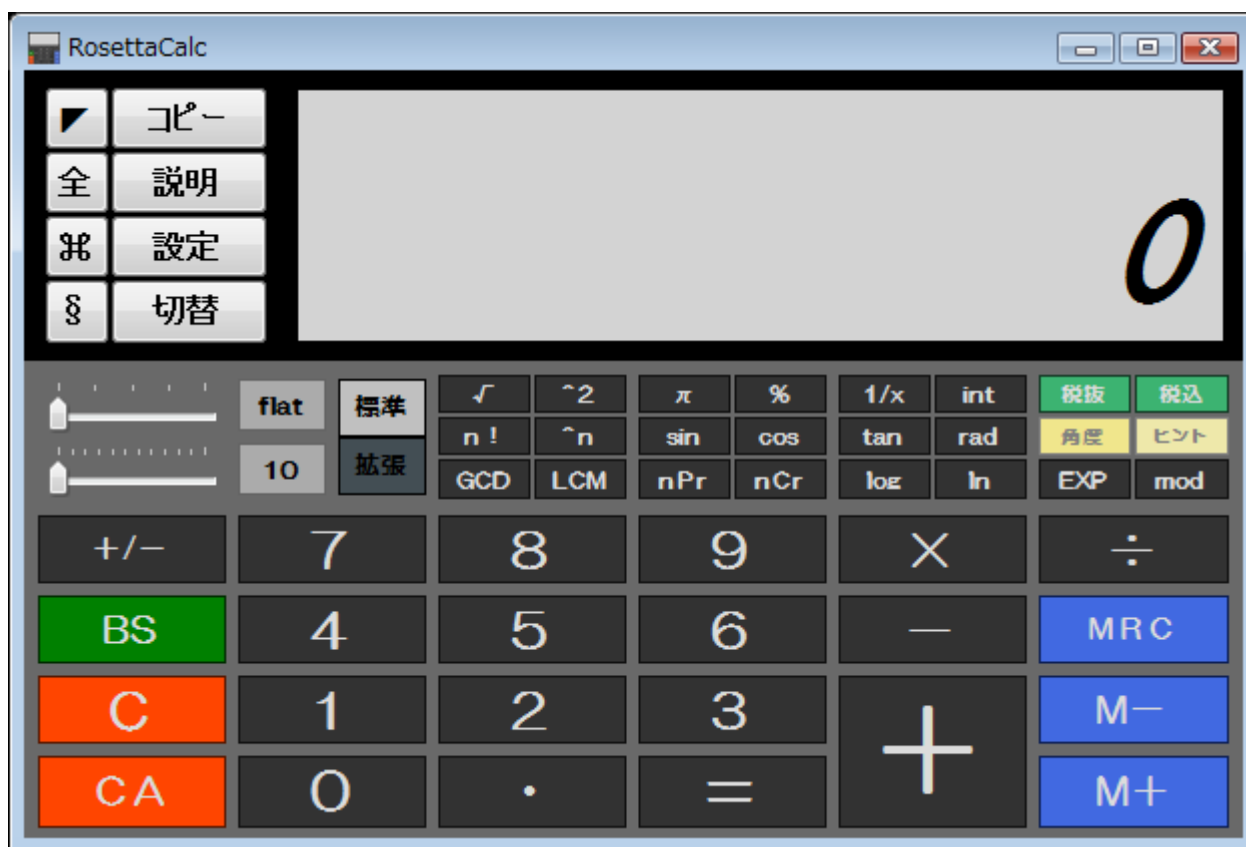


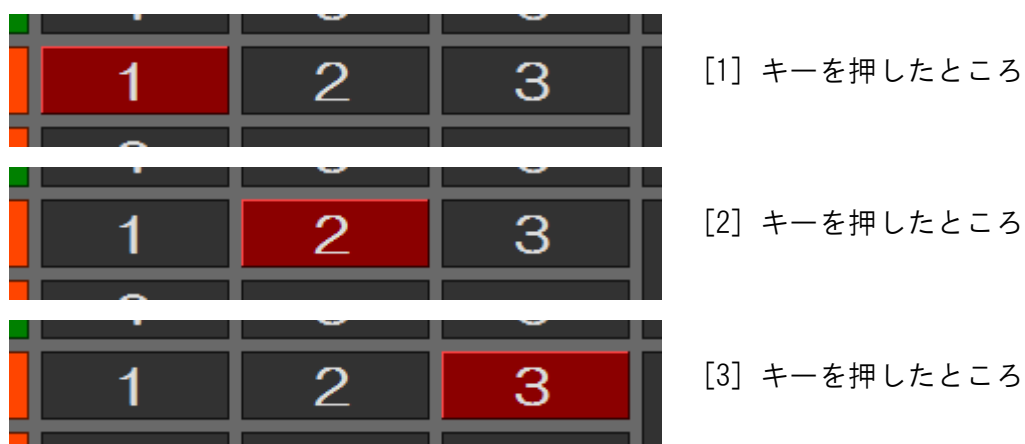
当アプリは電卓ソフトですので電卓画面が基本です。



(画像は Windows Vista で撮影しており、ご利用のOSによっては印象等が異なる場合があります。)

入力には、電卓画面に設置された数字キーのほか、お使いのキーボードのテンキーやキーボード上部の数字キー、または市販のテンキーパッドなども使えます。

数字キーや演算子などは、マウスオーバー（マウスのポインタが重なること）やキーボード等からの入力操作を受けた際には、該当するキーが一時的に「暗赤色」に点灯するようになっています。



キーボード等から入力可能かどうかは、電卓画面がアクティブ(最前面で動作中)かどうかで判断できます。この状態は、一般的には外枠部分や上部タイトルバーの表示状況などでもわかりますが、当アプリでは、電卓画面で広い面積を占める、表示域の「明るさ」でも判別できるようにしています。

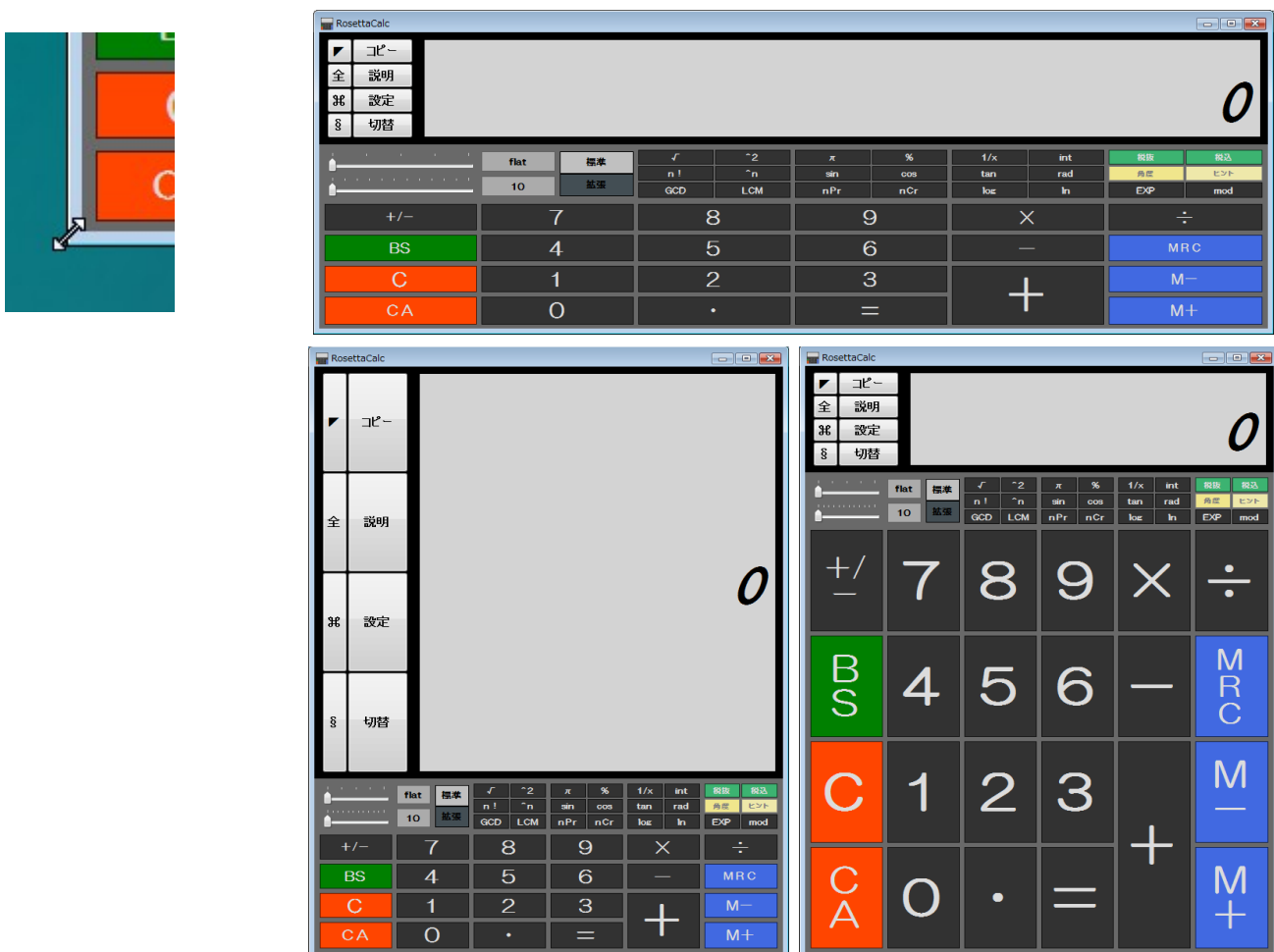


(左：非アクティブ=暗、右：アクティブ=明)

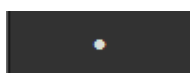
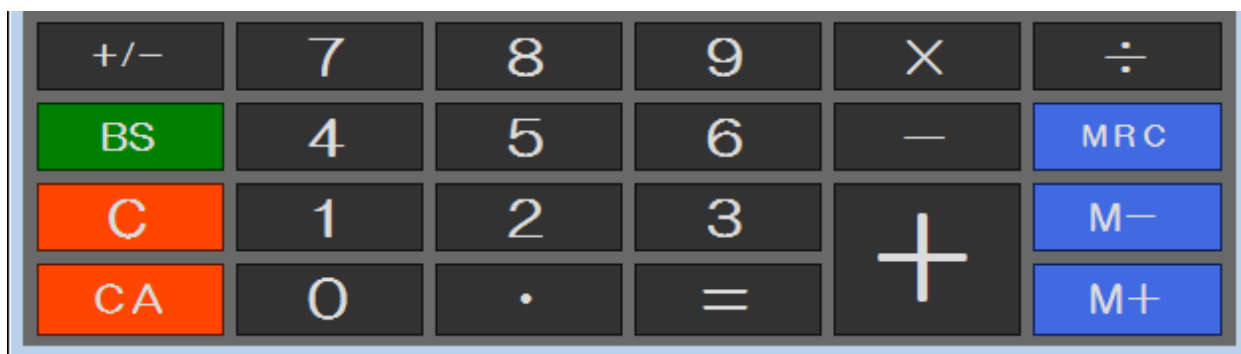
表示窓と操作部との間などには、マウスで幅を調節できる境界線があります(下図オレンジ色部分)。テンキー等を併用するのであれば、操作部をすべて隠して、表示窓だけに使うことも可能です。



電卓画面は、外枠部分をマウスでドラッグして、広げたり縮めたりできます。電卓に設置された数字キー等の表示は、キーの縦幅に応じて拡大縮小します。



電卓画面の下段部分は、一般的な電卓と同様で、数字キーや演算子などが集まっています。

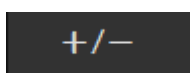


[.] 小数点 (表示の視認性を高めるため [.] (中黒) を用いています。) テンキー等の小数点でも、キーボードのピリオド(フルストップ)でも構いません。

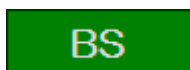


[=] イコール 四則演算などの結果を求めます。テンキーやキーボード等の [=] キーでも、[Enter (Return)] キーでも構いません。

テンキーやキーボード等では一般に、[×] や [÷] については [*] キーや [/] キーが対応します。もちろんそれらについても、当アプリでも同様の入力操作ができるようになっており一般の電卓と何ら変わらない操作性を持っています。このほか、機能的なものに説明を加えます。



[+/-] プラスマイナス 現在の数値の、正と負の符号を切り替えます。(明示的な符号はマイナスのみです。プラスの場合には付きません。[f12] も可)



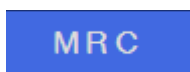
[BS] バックスペース 入力中の数値の末尾の数字1桁を削除します。([BackSpace(BS)] キー、[←(左向きの)delete] キーも可)



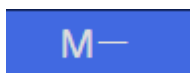
[C] クリア (消去) ([esc(Escape)] キー、[Clear] キーも可) (最後に演算キーを押したあとの) 入力中の数値をゼロにします。



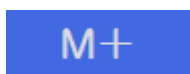
[CA] クリア・オール (全消去) (御破算: ごわさん) ([Delete] キーも可) 前の計算結果を含めた (メモリー機能以外の) 内容を、すべてクリアします。



[MRC] メモリー機能で記憶させた数値の「呼び出し」と「消去」をおこないます。1度押せば呼び出し、続けてもう一度押せば消去 (クリア) されます。([ctrl]+[M]、[ctrl]+[*] (Control キーを押しながら[M]または[*] キー) も可)



[M-] メモリー内の数値に対して、表示中の数値を (暗黙的に) 減算します。([ctrl]+[-] (Control キーを押しながらテンキー等の[-] キー) も可)



[M+] メモリー内の数値に対して、表示中の数値を (暗黙的に) 加算します。([ctrl]+[+] (Control キーを押しながらテンキー等の[+] キー) も可) (通常の利用では 初期値ゼロのメモリーに記憶させる最初のボタンになります。)

電卓画面のキートップに表示する、数字や一部の文字は、設定画面にて変更可能です。(文字の変更は表面上だけのもので、機能上や実質的な変化はありません)

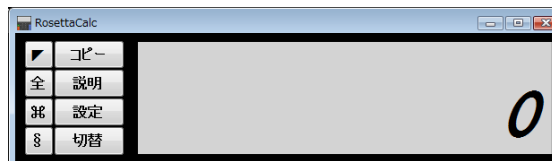
● 電卓機能

あらためて説明するまでもありませんが

計算を行う場合の、表示域の遷移状況は、次のようになります。

(「1 2 3 4 + 4 3 2 1 =」の計算例。数値表現は初期状態)

初期値 ([CA] を押す)

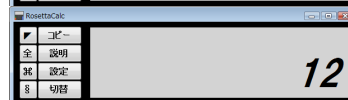


(必要な場合)

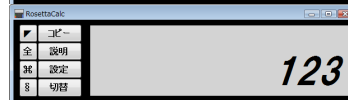
[1] を押す



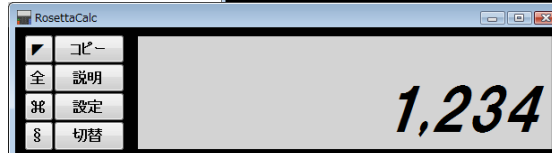
[2] を押す



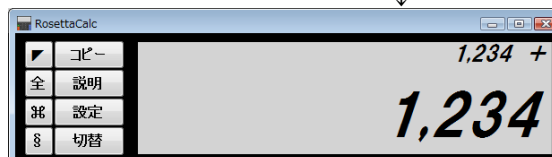
[3] を押す



[4] を押す



[+] キーを押す

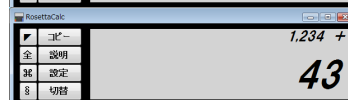


(右上に確定した数値と符号が表示)

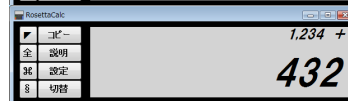
[4] を押す



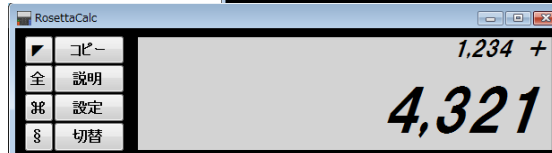
[3] を押す



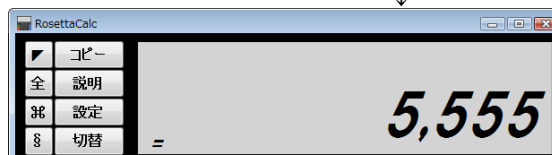
[2] を押す



[1] を押す



[=] キーを押す



(計算結果。左下に [=] と表示)

以上のように、一般の電卓とまったく同じ感覚で操作できます。

四則演算の [+][-][×][÷] のいずれも、同様の操作で計算できます。

計算の途中においても、右上部分に入力済の数値が

四則演算の符号とともに表示され続けるので、わかりやすくなっています。

● 連続計算機能

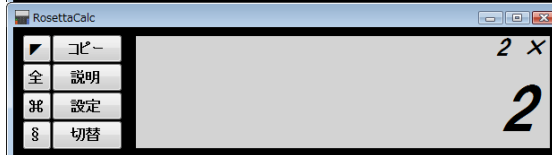
[$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \dots$] のように、同じ計算を何度も繰り返す場合には、次のようにもできます。

[2] を押す



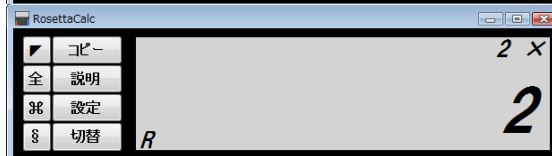
(唯一の基本数値)

[×] キーを押す



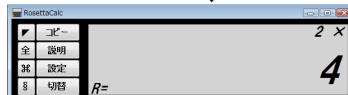
(右上に確定した数値と符号が表示)

再度 [×] キーを押す



(左下に [R] と表示)

[=] キーを押す



($[2^2] = 2 \times 2$ 。左下の表示は [R =])

[=] キーを押す



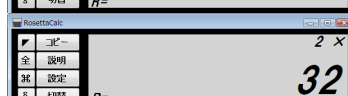
($[2^3] = 2 \times 2 \times 2$)

[=] キーを押す



($[2^4] = 2 \times 2 \times 2 \times 2$)

[=] キーを押す



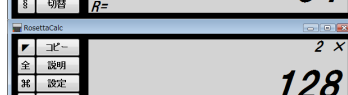
($[2^5] = 2 \times 2 \times 2 \times 2 \times 2$)

[=] キーを押す



($[2^6] = 2 \times 2 \times 2 \times 2 \times 2 \times 2$)

[=] キーを押す



($[2^7] = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$)

[=] キーを押す



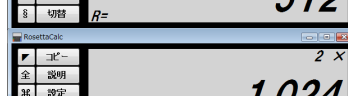
($[2^8] = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$)

[=] キーを押す



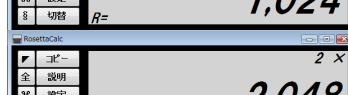
($[2^9] = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$)

[=] キーを押す



($[2^{10}] = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$)

[=] キーを押す



($[2^{11}] = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$)

四則演算の [+][−][×][÷] のいずれも、同様の操作で処理できます。

(上の例のように [2][×][×] で始めた場合は、[2ⁿ] の n を「 [=] の回数 (+1)」であらわしていることとなります。)

最初に入力する基本数値は、複数桁でも、小数部を含んでも構いません。

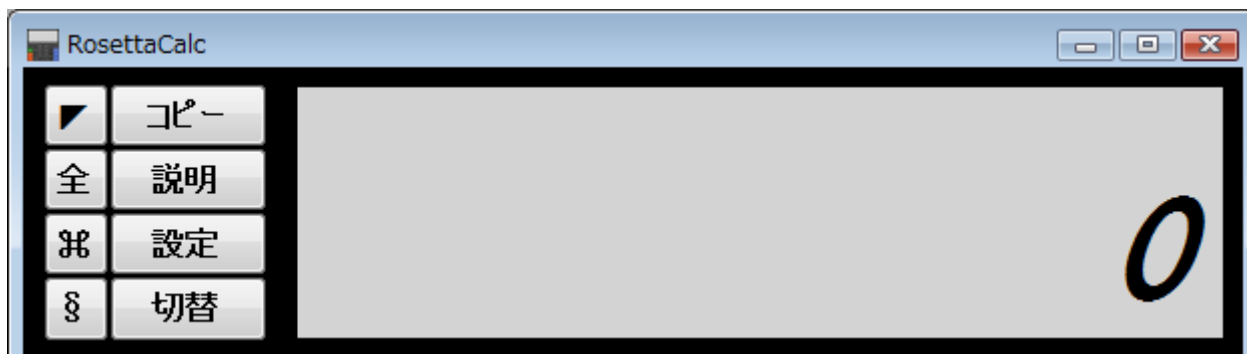
(上の例ではわかりやすくするため 1 桁の整数にしています。)

[=] キーの連打は「長押し」でも、キーボードの [Enter] (return) キーを押し続けても、同様になります。





(「長押し」などの場合は、計算が進行する速さや、最初に入力する基本数値の大きさなどにもご注意ください。)

※ [特定の数値] と [連続する数値] との四則演算などは「直算電卓」機能にて扱っています。

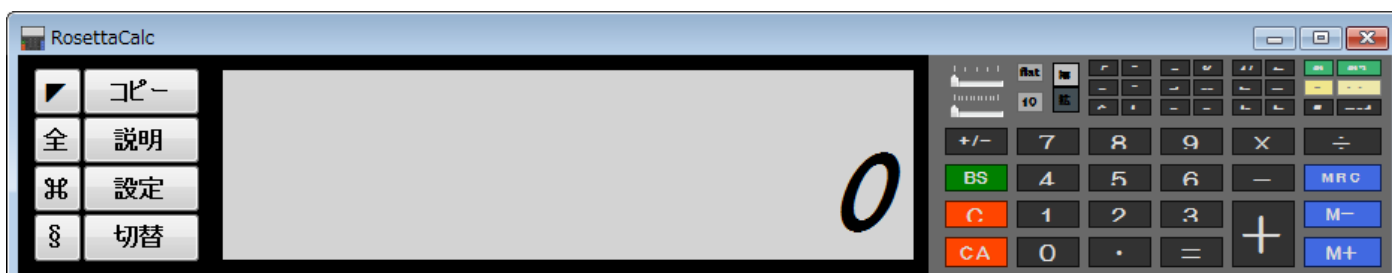
上段部分は、機能ボタンの操作部と、数値表示のための表示域とで構成されています。



表示域の数字は、視認性を高めた見やすい大きさにしており、調節もできます。

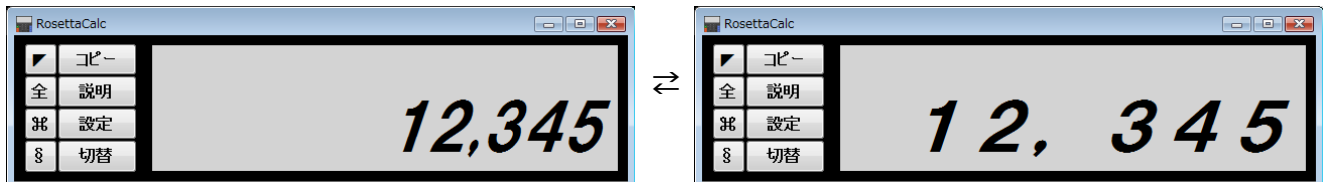
-  [] ボタンは、電卓画面のレイアウトを、縦型／横型で切り替えます。
-  [全] ボタンは、表示域の数字を「全角文字」に変換できるかどうか、試すことができます。
-  [⌘] ボタンは、通常の電卓画面と、当アプリ独自の「直算電卓」とを切り替えます。
-  [§] ボタンは、画面左上部分の一部の操作系を切り替えられます。

[] ボタンは、電卓画面のレイアウトを、縦型／横型で切り替えます。



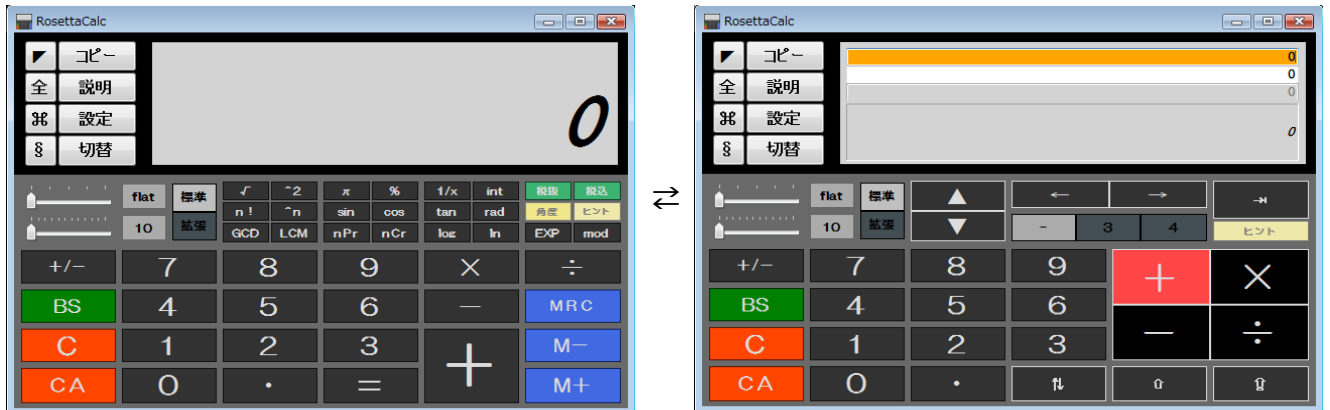
横型にすると、電卓画面の左上の位置を基準にして、黒枠の上段部（操作部と表示域）の大きさは変えずに中段部と下段部を、上段部の右側へ配置します。もう一度 [] ボタンを押すと、元に戻ります。

[全] ボタンは、表示域の数字を「全角文字」に変換できるかどうか、試すことができます。



全角/半角の区別のある、一般的な「アラビア数字」や、ラテン文字による「ローマ数字」などの場合を想定しています。変換できた場合は、もう一度押せば「半角文字」に戻せる場合もあります。「漢数字」などには非対応です。

[㊦] ボタンは、通常の電卓画面と、当アプリ独自の「直算電卓」とを切り替えます。（後述）



あらかじめ選んでおける四則演算の演算子と、入力可能な2つの数値を使って、入力中も直接計算します。片方の数値が一定な場合や、連続する数値をもとにした計算などには、特に効果を発揮します。（後述）

[§] ボタンは、画面左上部分の一部の操作系を切り替えられます。（もう一度押せば元に戻ります。）（設定画面の「各種機能」欄で「進法入力」をおこなうよう設定すると、[§] ボタンの切替は、3通りになります。）



表示域の数字の大きさは [Δ] / [▽] ボタンで調節できます。

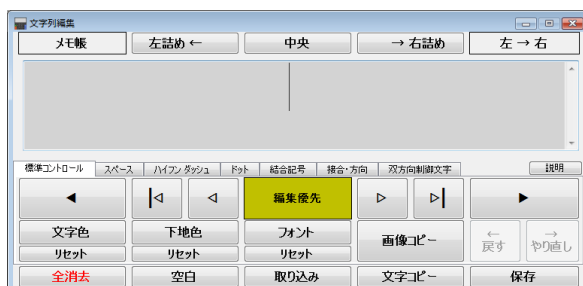
（初期値の大きさに戻すには [Δ] と [▽] の間にある細長いボタンを押します。）

- [alt] キーを押しながら押すと、入力中の数値の末尾桁（整数または小数）を1ずつ増減できます。

右上の[コンボボックス]では、表示域のフォントの種類を「一時的に」切替えます。

（継続的に変更する場合は、設定画面にて指定します。）

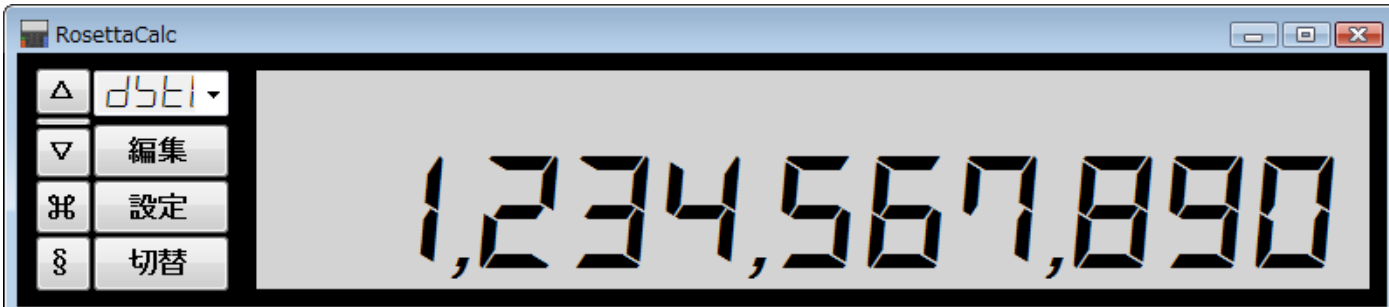
[編集] ボタンは、「文字列編集」画面を開きます。（下図；詳しくは別稿をご参照ください）



標準的な日本語環境ではあまり使われない、「EMスペース」「Three-Em Dash」「結合用ダブルオーバーライン」などの特殊な文字や、アラビア語・ヘブライ語・ディベヒ語（ターナ文字）などの「右横書き」環境、あるいは「左横書き」と「右横書き」とで書字方向が混在する場合（双方向制御）などの、文字列操作に関する補助的な機能を備え、表現の細部などへの調整に役立つことが期待できます。

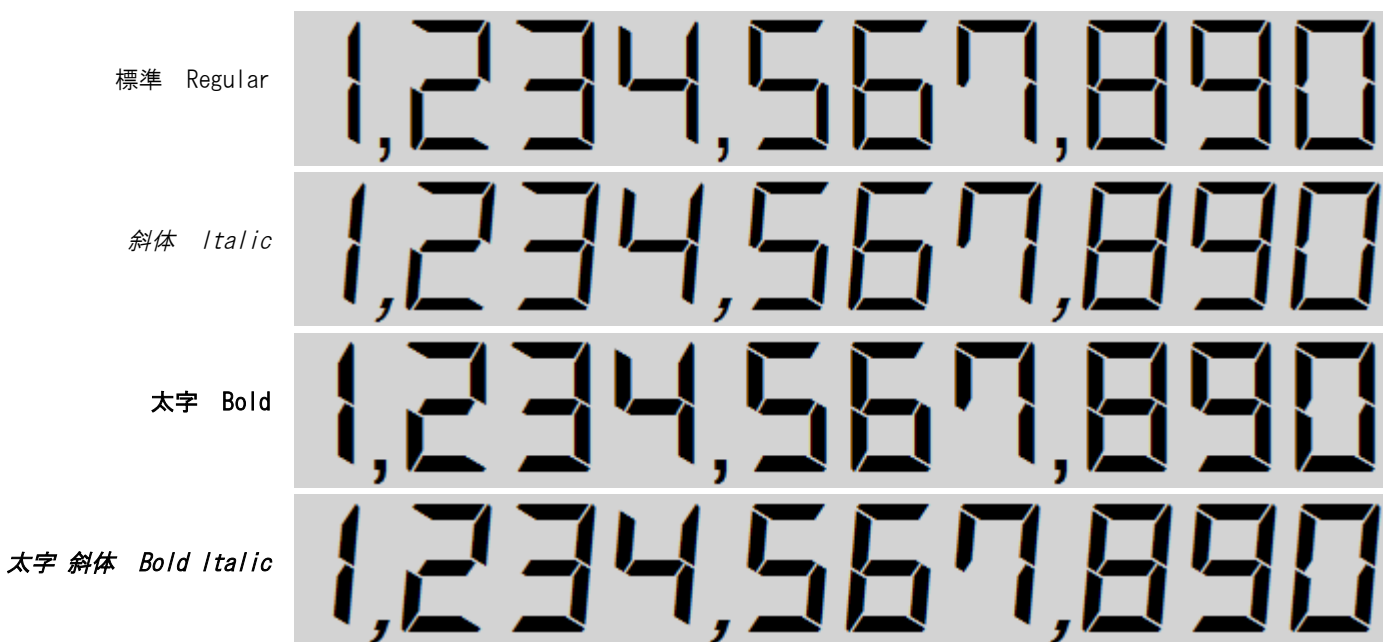
フォントを切り替えると、表現の幅を広げることができます。

たとえば、実際の電卓やデジタル時計などの液晶画面に見られる「セグメント表示」を模したフォントなども使えます。

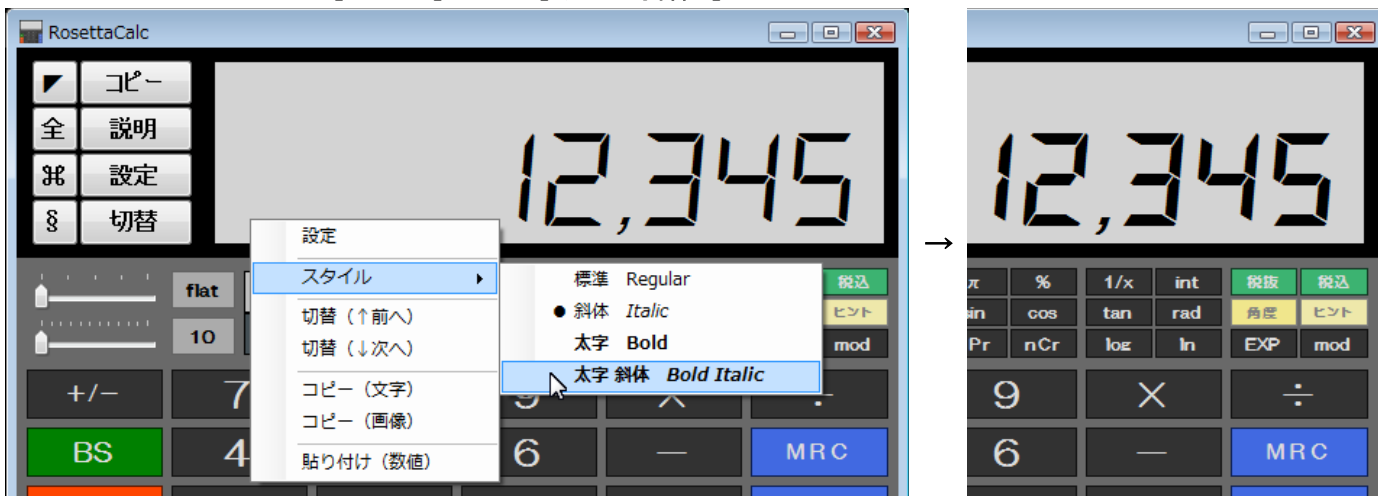


(画像の例は「けしかん」様作成の [DSEG7 Modern] より [*BoldItalic*] です。紙面の都合で画像は若干縮小しています。)

フォントのスタイルは、当アプリでは基本的に [標準] と [斜体] に対応しているフォントが対象です。表示域では [標準 Regular]、[斜体 *Italic*]、[太字 **Bold**]、[太字 斜体 *Bold Italic*] の最大4種類から選んで利用できるようになっています (可能な場合)。



スタイルは、表示域を「右クリック」すると表示される「スタイル」から切り替えることができます。次の画像は初期設定値の [斜体] から [太字 斜体] に変更しているところです。



通常のスタイルは、設定画面で選択した [斜体] もしくは [標準] になっています。

[太字] や [太字 斜体] への変更は、入力用の選択で計算を続ける場合など、フォントが切り替わらない間は、継続されます。(フォントを変更したり、「切替」ボタンで数値表現を切り替えた場合には、[斜体] または [標準] に戻ります。)

● 適切な表示のために

前ページの例は、次の図に示した [DSEG7 Modern] の4つのフォントファイルを、システムにインストールして試したものです。
(中央の画像は公式サイトから zip 圧縮ファイルをダウンロードして解凍したフォルダ内の状況(抜粋)です。)

フォントファイルの構成はそれぞれのフォントにより異なりますが、念のためもう少し例示しておきますと、このフォントで Regular だけをインストールした場合には、左側ようになります。(右側は前頁同様4つともインストールした場合です。)



[DSEG7 Modern] では、Regular を単体でインストールした時の 斜体 の大きな傾きは、Italic の追加で改善されるようです。また Regular だけで 太字 にした場合には、個々のセグメントの境界が潰れて不明瞭になっています。これらのことからこのフォントの場合は、個々のフォントファイルにより、細かな調整が反映されるように作成されているものと思われます。(一般のフォントでは、ここまでの細かな配慮はされておらず、上図の左側の仕組みだけで表示されていると想像できます。)

以上のように、適切に表示するには、思い込みを排除し、各フォントの仕様を(ある程度でいいので)理解することが必要です。

- ※ フォントは Windows だけでなく Mac や Linux など各種プラットフォームで利用可能なように作成されています。上記フォントの [-Light] や [-LightItalic] 等については、Windows 系アプリでは利用しづらい場合があります。
- ※ 当アプリを含む、多くの Windows 系アプリは、その起動時に、システム内のインストール済みフォントを認識します。新しく追加したばかりのフォントを使って表示する場合は、当アプリを再起動して(再認識させて)ご利用ください。
- ※ 当アプリは開発環境の仕様により、TrueType(. ttf) や OpenType(. otf) などのフォントには対応していますが [PostScript Outline] を用いたフォントについては「非対応」となっております。ご了承ください。

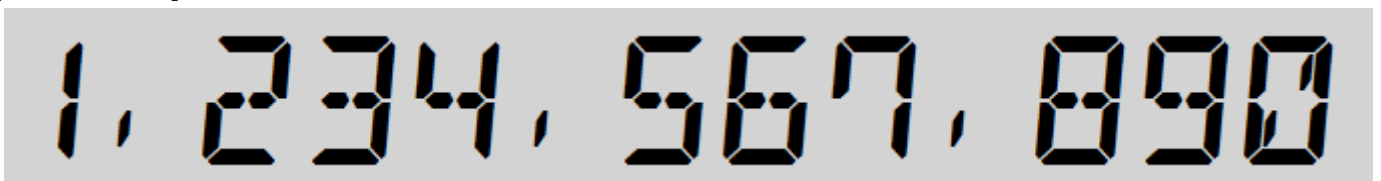
一般の電卓なら アルファベットは使わず ほぼ「アラビア数字」だけでいいのでフォントにはシンプルなものを選んだほうが、より明瞭になります。

参考のため、パッケージに入っている他の関連フォントについても、主なものを挙げておきます。(以下のスタイルはすべて [太字 斜体] にしています。)

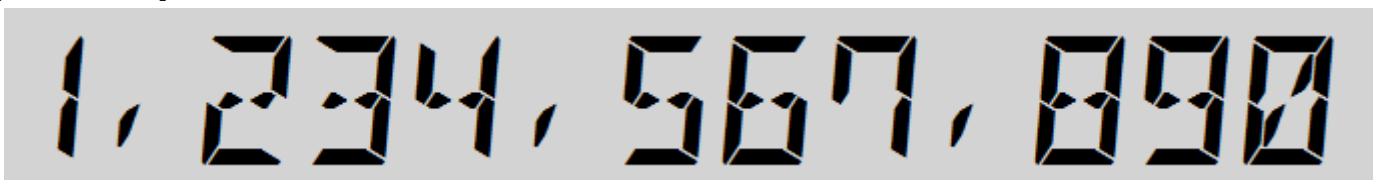
[DSEG7 Classic]



[DSEG14 Classic]



[DSEG14 Modern]





- [**コピー**] ボタンは、表示域の数値表現を、クリップボードにコピーします。
- [**説明**] ボタンは、各種の資料等をまとめた、PDF 表示用ランチャーを開きます。
- [**設定**] ボタンは、電卓の各種機能の設定をおこなう「設定画面」を開きます。
- [**切替**] ボタンは、表示域の数値表現の切り替えに使います。

[**コピー**] ボタンは、表示域の数値表現を、クリップボード（パソコンのコピー用領域）にコピーします。
 （当アプリの電卓の数値表現の文字列をコピーして、他のソフト等に「貼り付け」する際に使えます。）

貼り付けた際に、表示の状況が異なる場合は、貼り付けたソフト等の側で、フォントの種類などを揃えてください。
 電卓でコピーした内容が、他のソフトで「数値」として扱えるかどうかは、選択した数値表現の状況などによります。
 たとえば「ローマ数字」や「漢数字」などの「文字」による数値表現を「数値」として認識してくれるソフトは、かなり少数派ですが、一般的な「アラビア数字」なら、表計算ソフト等でも「数値」として扱ってくれることが多くなります。

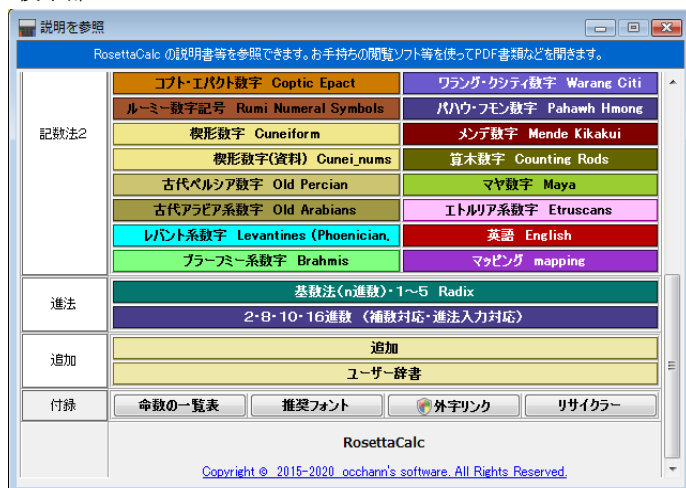
※ 数値表現を文字列としてコピーする機能は、キーボードの [ctrl]+[C] でも利用できます。

[**説明**] ボタンは、各種の資料等をまとめた、PDF 表示用ランチャー（「説明を参照」画面）を開きます。

前半部



後半部



個々の PDF 説明書は、ご利用の「PDF 閲覧用ソフト」を使って開きます。
 （PDF 閲覧用ソフトとは、Windows 標準搭載の「Adobe Acrobat Reader」などのことです。）
 （海外製ですが日本語対応のフリーソフト「PDF-XChange Viewer」も とても便利です。）

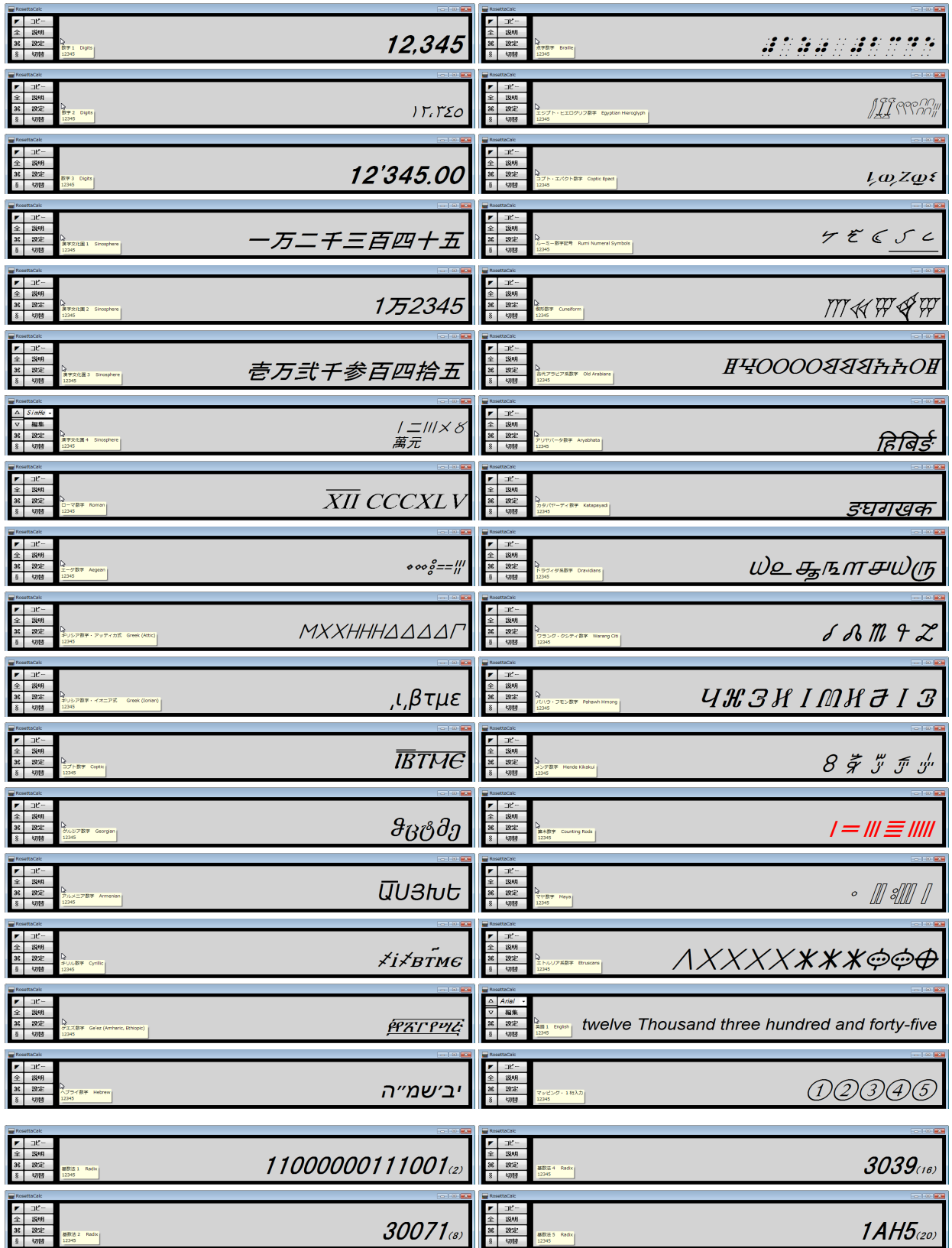
各種 PDF 説明書のうち、数値表現などについての大部分は、設定画面からも開くことができます。
 これらは色分けにより区別できるようにしており、設定画面でのページ順に区別しています。

いくつかの PDF 説明書は、執筆途中のためリリースが遅れるものがあります。
 それらが当方サイトにリリースされたあとに当アプリから説明書を参照すると、当方サイトの説明書を参照でき
 またご利用の PDF 閲覧用ソフトで開けるよう、自動的に所定の場所へダウンロード・インストールすることも選べます。
 同様に、PDF 説明書が更新される場合に備えて、上記 PDF 表示用ランチャー（もしくは設定画面）の各ボタンを
 「shift キーを押しながら」クリックすると、当方サイトにリリースされている最新版 PDF を確認でき、必要に応じて
 自動的に所定の場所へダウンロード・インストールすることも選べます。

後半部の末尾に「付録」としている [命数の一覧表] [推奨フォント] [外字リンク] [リサイクラー] の 4 項目は
 各ボタンをクリックするとそれぞれ付属のアプリ等が起動するようになっていきますので、必要に応じてご利用ください。
 （詳細は後述しています。）

[設定] ボタンは、電卓の各種機能の設定をおこなう「設定画面」を開きます。（後述）

[切替] ボタンは、当アプリ最大の特徴となっている、表示域の数値表現の切り替えに使用します。
 （キーボードの [Page Down] / [Page UP] キーでも切り替えでき、途中でも [Home] キーで入力用の表現に戻せます。）



（上記はすべて[12345]の数値表現です(抜粋)。上記のうちいくつかは、例示用に設定を変更しています。）

（楔形数字は[六十進法]($12345 = 3 \times 60^2 + 25 \times 60 + 45$)です。）

（マヤ数字は上図右下の20進数[1AH5]と同様の[二十進法]($12345 = 1 \times 20^3 + 10 \times 20^2 + 17 \times 20 + 5$)です。）

[設定] ボタンは、電卓の各種機能の設定をおこなう「設定画面」を開きます。
 (ここでは概要を記します。詳しくは別稿「設定画面」の説明をご参照ください。)



電卓画面の表示域での、基本となる表示スタイル ([標準] (正体) か [斜体] か) は基本設定の右上にある「電卓の表示スタイル」欄で選択します。

これはあくまで基本の設定です。上図では初期値の [斜体] が選択されています。

電卓の表示域では [標準 Regular]、[斜体 Italic]、[太字 Bold]、[太字 斜体 Bold Italic] の最大4種類から選んで利用できるようになっています。(後述)

数は少ないですが、フォントの種類によっては[太字]しかスタイルを持たないようなフォントもあります。基本設定に[標準]と[斜体]しかないのは、当アプリでは[太字]だけのフォントは扱わないことも示しています。

電卓画面の下段部分に配置されている、数字キーなどのキートップに表示される数字の文字は「電卓の表示スタイル」欄の下側にある「キートップの変更」ボタンから開く画面にて変更できます。

表示域の文字色と下地色、および、電卓本体の下地色と数値ボタンの色については2色のペアで指定するようになっています。

上図では上段の「表示域」欄に、初期値の **基本色** (文字色=黒、下地色=明灰色) が選択されています。その下にある **白黒** や **黒白** にも切り替えでき、あるいは中央寄りの **[変更]** を選ぶと、好みの色に変更できます。

また「本体色」欄では、初期値として、左上隅の暗灰色をベースとした「シック」が選択されています。ここでは計12色のプリセットカラーが選択でき、あるいは中央寄りの **[変更]** を選ぶと、好みの色に変更できます。

2色のペアは、ともに **[変更]** の上の2色の箇所をクリックすると、色の設定画面が表示され、好みの色に変更できます。

数値表現に適したフォントは、「表示切替」の各行にある「フォント」ボタンから設定できます。

通常使うフォントの種類は、「入力用」として選択する「数値表現のひとつ」と捉えて指定します。

上図では下段「表示切替」欄の数値表現のうち「数字・1」が「入力用」として選択されています。通常使うフォントは、「入力用」に選んだ行の「フォント」ボタンから、切り替えにより指定できます。

電卓画面の [税抜][税込] ボタンによる計算の基準とする税率は、「税計算」欄の「税率」で設定します。

初期値では、日本の消費税率（本稿執筆時点(2018年)で現行の8%）に設定されています。

将来的な法改正による税率変更には、当アプリではバージョンアップ（クリーンインストール）でも対応予定ですが本項の税率をご自分で直接変更することでも対応可能です。

日本以外での税額計算にも対応可能なよう、税率は念のため小数第二位まで設けていますが計算手順まで同じとは限りませんので、海外向けにお使いの場合は、別項に記載の計算手順（前述）をご理解のうえ利用の可否をご判断いただきますようお願いいたします。

電卓画面左側の、2段目のスライダを右端にした場合にのみ適用される、多桁計算用的小数部の桁数は「小数点以下(拡張)」欄の「有効桁数(予備)」と「最大桁数」の組み合わせにより設定できます。

2段目のスライダを右端にして電卓画面に明示されるのは、「有効桁数(予備)」で設定した桁数です。

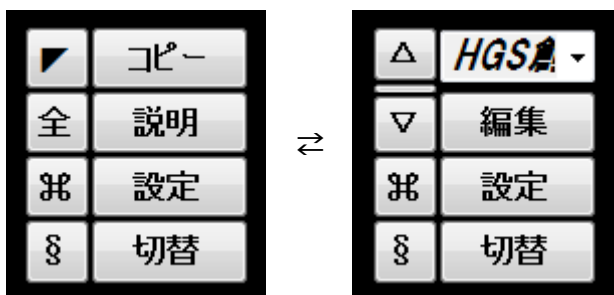
この桁数は1段目のスライダで [flat] (端数処理なし) 以外を選んだ場合におこなわれる「端数処理」に適用されます。

「最大桁数」は、1段目のスライダで [flat] (端数処理なし) を選んだ場合に、あたかも無制限のようになる小数部の現実的な計算範囲を決めると同時に、「有効桁数(予備)」の設定可能な範囲を決める、ストッパーとしても機能します。

「各種機能」欄で「進法入力」をおこなうよう設定すると、[§] ボタンの切替は、3通りになります。

[進法入力] は、通常の [10進数] とは異なる [2進数][8進数][16進数] の形式で、入力や計算をおこなうものです。2進数なら [0] と [1] のみなど、入力に用いる数字キーの個数が異なるため、それに対応するための選択肢が追加されます。

進法入力 (チェックなし・「 § 」ボタンによる遷移は2通り)



進法入力 (チェックあり・「 § 」ボタンによる遷移は3通り)



変更した設定内容を電卓に反映させるには、設定画面の最下部にある「OK」ボタンを押します。

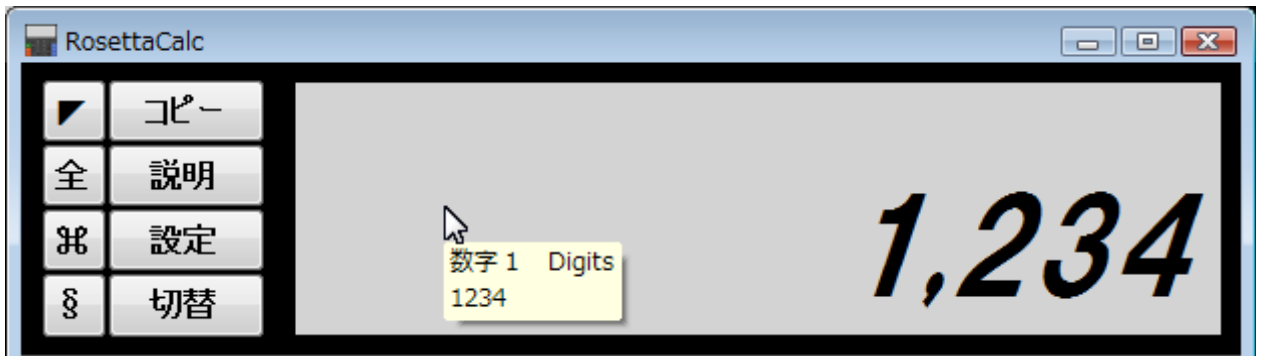
(OKボタンが押されない限り、電卓には反映されないようになっています。)



※ 設定画面については、別稿に詳しい説明がありますので、ご参照ください。

上段部分にある表示域の数字は、視認性を高めた見やすい大きさにしており、調節もできます。

画像の例での表示域のフォントは、初期設定の「HGS 創英角ゴシックUB」（斜体）です。



入力中の数値は、表示域にマウスのポインタを重ねれば、ツールチップ内にも表示されます。

表示域の数字は、電卓らしく見えるよう、初期設定では [斜体] にしています。

[標準] (正体) にする場合や、文字色と下地色の変更、さらに通常使うフォントの種類などは電卓の [設定] ボタンから開く「設定画面」にて指定できます。

数値の桁数が多くなり、表示域に表示しきれなくなると、数値の表示位置が、やや上に上がります。

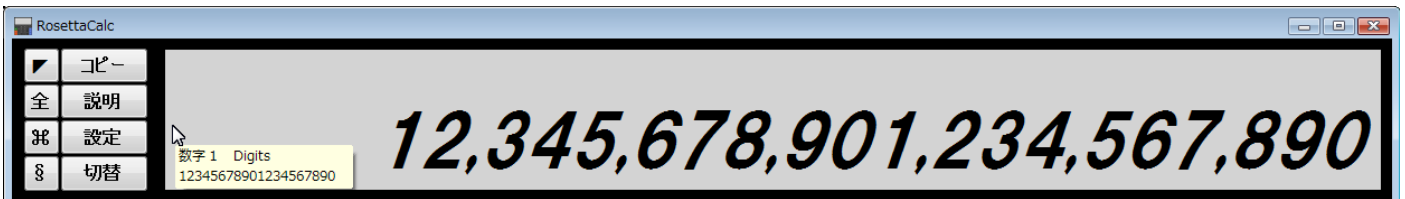
表示しきれているか、上下の位置だけではわかりにくい場合は、ツールチップに表示された数値と比べることも確認できます。



画像では、左側の 9 桁 [123, 456, 789] は表示できていますが、右側の 10 桁 [1, 234, 567, 890] は表示しきれず上に上がっています。

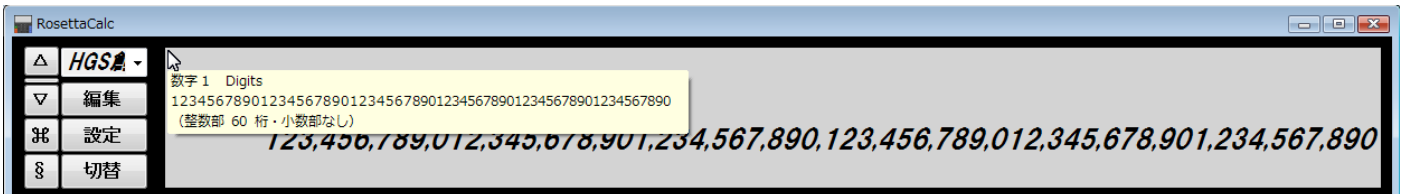
表示域の上辺や左辺には、計算途中の数値や状況などを表示する余白部分があるので、数値の表示部分はやや狭くなっています。電卓の数値表示は「右詰め」が基本ですが、桁数が多い場合は先頭側（左上）から詰まっているので、このような表示になります。（ここでは見えていませんが下に示すように、実際は 2 行に分かれているので、やや上に上がったような格好になっています。）

このような場合には、電卓の外枠をドラッグして電卓画面の幅を広げるか、表示域の文字の大きさを小さくしてください。

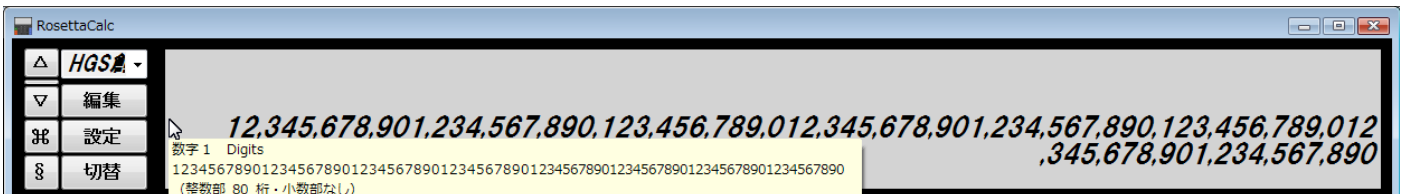


画像では横方向にのみ広げていますが、縦方向に広げることも、周囲の境界線を調節して全画面表示にすることもできます。

次の画像は、さらに [§] ボタンで操作部を切り替え、[▽] で文字の大きさを小さくして、入力数値を追加したところです。



文字の大きさが表示域の高さに対して十分に小さい場合には、数値の桁数によっては、複数行に渡って表示されることもあります。



数値は「右詰め」が基本ですが、桁数が多いため先頭側が詰まっており、表示域の幅で押し出された結果、2 行になっています。これは実際に数値の途中で改行されているのではなく、見かけ上の改行に過ぎません。見かけの改行位置は表示域の幅によります。

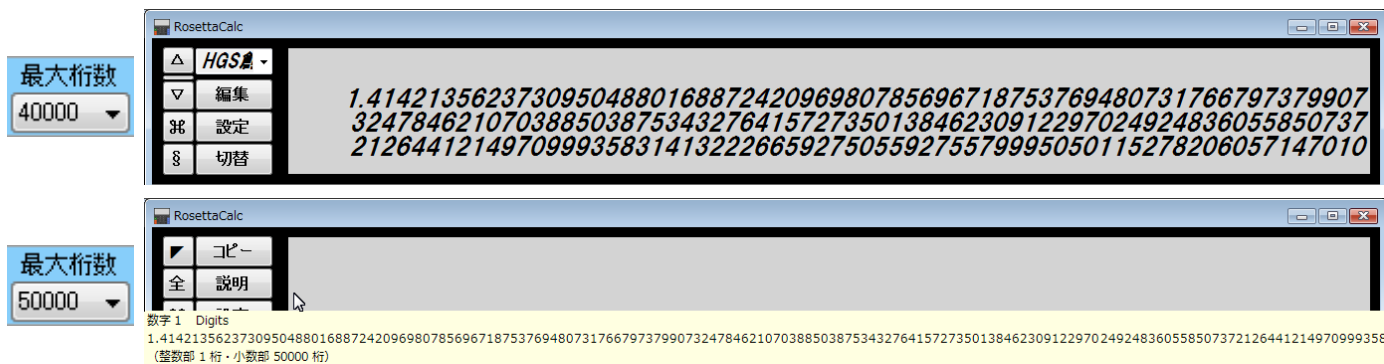
なお、表示域には「表示可能な文字数」が決まっています。

したがって（いずれにせよ）あまりにも大きな桁数の数値については「扱いにくい」ことは確かです。

大量の文字をまとめて表示するには、文字の大きさと文字数に応じた「面積」が必要です。

「表示可能な文字数」を超えている場合には、表示域での「表示」自体がおこなわれなくなります。

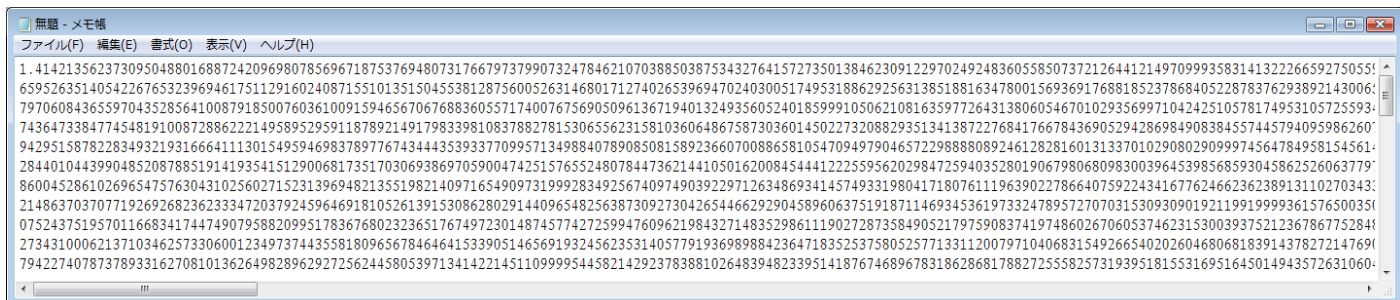
次の例では、上段側で小数部の最大桁数 4 万桁、下段側で 5 万桁の設定で、どちらも「ルート 2」を計算して試しています。



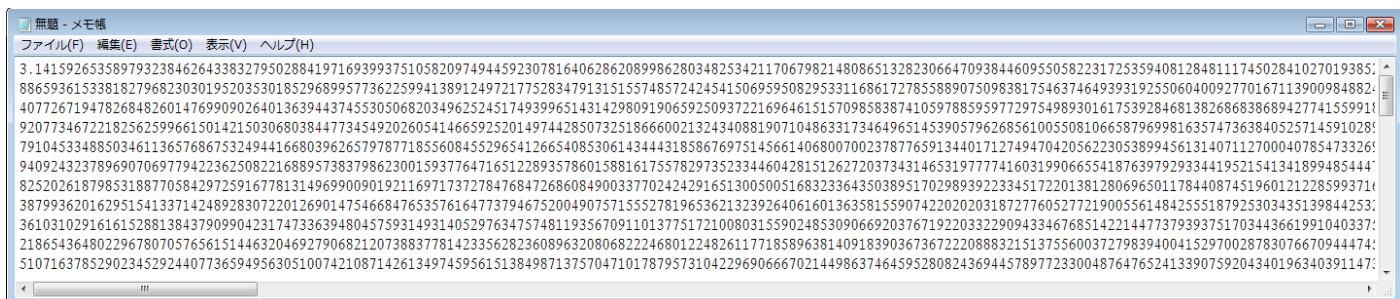
小数部 4 万桁の場合は表示できていますが、5 万桁では一部がツールチップに載るだけで「表示」はされていません。（当方の調査では、表示域 (Label) の「表示可能文字数」は、非サロゲートペアで [43680 文字] のようです。）

上段側のように「なんとか表示できている」状態では、表示にかかる負担が大きく、かえって動作等に悪影響が及びます。下段側のように「表示されなくなった」状態のほうが、動作はむしろスムーズになります。

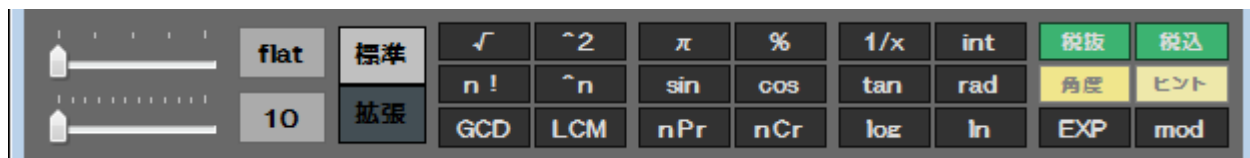
表示域には「表示」できなくとも、ツールチップにあるように電卓内での「計算」はできていますので「コピー」ボタンや [ctrl]+[C]、または表示域を右クリックして表示されるメニューからの「コピー」で数値の文字列をコピーして、「メモ帳」などの別アプリに「貼り付け」すれば文字列としての利用は可能になります。



このような使い方が必要になるのは「平方根」や「円周率」など特定のケースに限られると思われそうですが、当アプリではとりあえず「やってみればできる」ようにしていますので、お試しください。



中段部分も 上下にある境界線で表示幅を調節して隠してしまふことができます。（お子様の利用時など）



中段左側には、除算（割り算）などで発生する小数部の処理方法（端数処理）や多桁計算（多倍長演算）に対応する場合の計算方法（型）を指定できる、操作部分があります。

切り上げ・切り捨て・四捨五入などの「端数処理」が必要な場合は、上段側のスライダで選択できます。端数を処理した結果として求める「小数部の有効桁数」は、下段側のスライダで指定します。多桁計算をおこなう場合の計算の「型」は、[標準] [拡張] の切替ボタンで選択します。

端数処理は「丸め」ともいいます。初期値では [flat]（端数処理なし）が選択されています。整数部における任意の桁位置での端数処理については、当アプリでは対応しておりません。

上段側の「端数処理スライダ」は、左から順に

- [flat] 端数処理なし （端数のことは一切考慮せずに計算します。下段スライダの桁数は無視されます。）
- [CUT] 切り捨て （指定した桁数になるよう「切り捨て」します。）
- [UP] 切り上げ （指定した桁数になるよう「切り上げ」します。）
- [5/4] 四捨五入 （指定した桁数になるよう「四捨五入」します。もっとも一般的な丸め方です。）
- [5/5] 五捨五入 （四捨五入とほぼ同様ですが、四捨五入を一歩進めて欠点を減らした方法です。）

となっており、[flat] 以外の場合が、実際の「端数処理」となります。

左右に動かして、用途に適した方法をお選びください。

端数処理の対象が、「中間値(0.5)ちょうど」の場合は「ちょうど真ん中」なので、本来は「どちらでもよい」はずですが「四捨五入」の場合は「五入＝必ず切り上げる」ことになるので、回数を重ねると「切り上げ側へ傾いている」ようになります。これを問題視する場合、本来の「どちらでもよい」でも困るので、「どちらかに振り分けよう」とするのが「五捨五入」です。つまり問題となるのは「中間値ちょうど」の場合のみで、それ以外は四捨五入と同じです。（詳しくは後述しています。）

下段側の「有効桁数スライダ」は、端数処理によって求める、小数点以下の桁数を決めるものです。

有効桁数は、上段の端数処理スライダが [flat] の場合は無視されます。それ以外の場合はすべて「端数処理」をおこないますので、その際的小数部をあらわす桁数に用いられます。

左から [10]～[0] と、連続的に切り替えできます。また、右端にしたときの初期値は [20] です。

右端の位置は、小数部についての「多桁計算」に対応します。

右端の位置での具体的な桁数は、設定画面にて変更できます。（0～10万桁）

たとえば、[flat] 以外で各種「端数処理」がおこなわれる場合は、次のようになります。

- [2] なら、求める小数部は2桁です。小数第三位を端数として処理し、**小数第二位** に丸めます。
- [1] なら、求める小数部は1桁です。小数第二位を端数として処理し、**小数第一位** に丸めます。
- [0] なら、求める小数部は0桁です。小数第一位を端数として処理し、**小数部のない 整数** に丸めます。

[flat] 以外の「端数処理をおこなう選択」になっている場合は、計算結果の小数点以下を、指定した桁数に揃えます。

[flat]（端数処理なし）の場合は前述の通り、下段側のスライダで設定した有効桁数の値は、無視されます。

[flat]（端数処理なし）の場合の小数部の桁数は、割り切れない場合などには、本来なら無限に続くため計算も終わらないので現実には実用に支障がない程度に制限されます。この桁数は明示されませんが、計算方法の選択により、次のようになります。

- 「標準」の場合は、計算に用いている「Decimal型」の限界値（小数部は32ビットOSなら多くても30桁程度）まで。
- 「拡張」の場合は、設定画面にて「有効桁数」のストッパー役として選択された「最大桁数」(+1桁)までの切り捨て。

[標準] [拡張] の切替ボタンは、数値の「計算」等に用いる「型」を切り替えます。

「多桁計算」（多倍長演算）の場合は、計算の型を [標準] から [拡張] へ切り替えて使います。（詳しくは後述しています。）

中段の関数ボタンには、次の種類を用意しています。（以下は一覧です。説明や計算例は次頁以降にあります）

√ 平方根	^2 二乗	π 円周率	% 百分率	1/x 逆数	int 整数化	税抜 税抜	税込 税込
n! 階乗	^n 冪(べき) (累乗とも)	sin 三角関数 サイン (正弦)	cos 三角関数 コサイン (余弦)	tan 三角関数 タンジェント (正接)	rad ラジアン/ deg 角度	角度 [角度]/ [ラジアン] 切替	ヒント ツールチップ ヒントの有無
GCD 最大公約数	LCM 最小公倍数	nPr 順列	nCr 組合せ	log 対数	ln 自然対数	EXP 指数関数	mod 剰余(余り)

消費税などの「税抜」「税込」といった計算は、緑色のボタンを押すだけで済みます。（税率は設定画面にて変更できます。）

(1行目: [f1]~[f8])

- √** **平方根** 数値Aの平方根(二乗すると元の数値に戻る値)を求めます。(使い方: [A] **√**)
- ^2** **二乗** 数値Aを二乗した値(2回掛けた値。自乗、平方)を求めます。(使い方: [A] **^2**)
- π** **円周率** 円周率(3.1415926...). π=パイ)の近似値を求めます。(使い方: **π**)
- %** **百分率** 元の値に百分率を掛けた値で計算します。(使い方: 四則演算の [=] の箇所 **%** を押す。)
- 1/x** **逆数** 1を元の数値Aで割って、元の数値Aの逆数を求めます。(使い方: [A] **1/x**)
- int** **整数化** 数値Aの小数部を取り除いた整数値にします。(使い方: [A] **int**)
- 税抜** **税抜** もとの数値を「税込額」とし、税率を用いて「税抜額」を求めます。(使い方: [A] **税抜**)
- 税込** **税込** もとの数値を「税抜額」とし、税率を用いて「税込額」を求めます。(使い方: [A] **税込**)

√ (square route) 平方 (square) π (pi) % (per-cent) int (integer)

(2行目: [control]+[f1]~[f8])

- n!** **階乗** 1から数値Aまでの整数をすべて掛け合わせた値(積)を求めます。(使い方: [A] **n!**)
- ^n** **冪(累乗)** 数値Aどうしの掛け算をB回分繰り返した値(A^B)を求めます。(使い方: [A] **^n** [B] [=])
- sin** **三角関数:サイン(正弦)** 直角三角形の内角Aに対する正弦値(sinθ)を求めます。(使い方: [A] **sin**)
- cos** **三角関数:コサイン(余弦)** " " " 余弦値(cosθ) " (使い方: [A] **cos**)
- tan** **三角関数:タンジェント(正接)** " " " 正接値(tanθ) " (使い方: [A] **tan**)
- a **rad** **ラジアン** 角度としての数値Aを [ラジアン] に直します。(使い方: [A] **rad**)
- b **deg** **角度** ラジアンとしての数値Aを [角度] に直します。(使い方: [A] **deg**)
- 角度** [角度]/[ラジアン]切替 上記 3. [sin] ~ 6. [rad]/[deg] の数値が [角度]か[ラジアン]か を選んでおきます。
- ヒント** ツールチップヒントの有無 マウスオーバーで表示される ツールチップヒントの 表示/非表示 を切り替えます。

sin (sine) cos (cosine) tan (tangent) rad (radian) deg (degree) ※ 角度は度分秒ではなく十進表記のみ対応

(3行目: [Alt]+[f1]~[f8])

- GCD** **最大公約数** 2つの数値の 最大公約数 を求めます。(使い方: [A] **GCD** [B] [=])
- LCM** **最小公倍数** 2つの数値の 最小公倍数 を求めます。(使い方: [A] **LCM** [B] [=])
- nPr** **順列** n個からr個を取る際の並べ方(順列)の総数Pを求めます。(使い方: [n] **nPr** [r] [=])
- nCr** **組合せ** n個からr個の取り出し方(組合せ)の総数Cを求めます。(使い方: [n] **nCr** [r] [=])
- log** **対数** Aを底とする数値Bの対数を求めます。(使い方: [A] **log** [B] [=])
- ln** **自然対数** 数値Aの自然対数を求めます。(使い方: [A] **ln**)
- EXP** **指数関数** 数値Aの指数関数(自然指数関数)を求めます。(使い方: [A] **EXP**)
- mod** **剰余** 2つの数値で除算(割り算)した[余り]を求めます。(使い方: [A] **mod** [B] [=])

G C D (Greatest Common Divisor) L C M (Least Common Multiple) P (permutation) C (combination)
log (logarithm) ln (natural logarithm) E X P (exponential function) mod (剰余演算子: modulus operator)

備考：「四捨五入」「五捨五入」について

端数を「四捨五入」する、という丸め方は、世界的にも広く用いられる方法で、問題視されることはまずありませんがよく考えてみれば「中間値(0.5)ちょうど」は、本来「どちらでもない」のに「必ず切り上げる」ことになっています。

この回数が多い場合、結果を集計したり平均を出したりすると「切り上げ側に傾いている」ということになるため要素の集計値や平均値としての正確性を期する場合などには「五捨五入」という方法が採られることがあります。

「五捨五入」というのは「中間値となった端数については最も近い偶数となる側へ丸める」という特殊な丸め方です。別名に「偶数丸め」「偶数への丸め」「最近接偶数への丸め」「最近接丸め」「銀行丸め」「偶捨奇入」などがあり英語でも「銀行家の丸め」(Banker's Rounding)といった呼び方があります。さらには、国際規格 ISO の [ISO 31-0] や日本国内向け JIS 規格 [JIS Z 8401] にて規格化もされていることから「ISO 丸め」「JIS 丸め」とも呼ばれるものです。

四捨五入の欠点を補うために使われることもあります。しかし、だからといって「正確だ」といえるものでもありません。というのは、どちらも「丸めている」ことに違いはないからです。端数処理して丸めた時点で「正確さ」は失われます。

正確性を重視する場合には、端数とする対象の桁位置を深くとるなど、影響を最小化するような対応をすべきです。少なくとも途中の計算においてはできるだけ、結果に影響を与えるような加工をおこなうべきではありません。

端数処理の対象とは「今まさに無視するかどうかを決めようとしている端数」であり、四捨五入が問題視されるのは「端数が中間値ちょうどになる回数が多い」「集計値や平均値を出すための要素の計算」といった場合に限られます。

一方、電卓の一般的な用途としては「その場限りの計算」が圧倒的に多いものと思われれます。そういった場合には特殊な「五捨五入」をわざわざ選ぶよりも、明快な「四捨五入」を選んでおくのも、良識ある方法かもしれません。

小数第二位に丸める場合 (小数部[2]桁)		
元の値	4/5 四捨五入	5/5 五捨五入
0.01499	0.01	0.01
0.015	0.02	0.02
0.01501	0.02	0.02
0.02499	0.02	0.02
0.025	0.03	0.02
0.02501	0.03	0.03

四捨五入では端数の対象桁に注目して「4か5か」だけで処理しますが「五捨五入」では、端数が「中間値(0.5)ちょうど」になる場合については丸めた結果が偶数になるように配慮した処理をおこないます。

端的に書き直しますと、端数が「中間値(0.5)ちょうど」の場合に限っては
 前の桁が**奇数**なら中間値(0.5)を**切り上げ** (←五入:四捨五入と同じ)
 前の桁が**偶数**なら中間値(0.5)を**切り捨て** (←五捨:四捨五入と異なる)となります。「中間値(0.5)ちょうど」でない場合は、四捨五入と同じです。

例示では黄色で示した箇所の端数が「中間値ちょうど」になっており赤色で示した箇所だけが、四捨五入とは異なる結果(五捨)になっています。

下の表は比較のために要素数を増やして集計してみたもので四捨五入では切り上げ側に接近(例示は極端で一致)しているのに対し五捨五入では平準化されて商の集計値に近づいていることがわかります。

ただし正確性を求めるなら余計な加工がない「端数処理なし」が最善です。

被除数	除数	商 (元の値)	flat 端数処理なし	整数値に丸める場合 (小数部[0]桁)				
				CUT 切り捨て	UP 切り上げ	4/5 四捨五入	5/5 五捨五入	
10	2	5.0	5	5	5	5	5	
9	2	4.5	4.5	4	5	5	4	
8	2	4.0	4	4	4	4	4	
7	2	3.5	3.5	3	4	4	4	
6	2	3.0	3	3	3	3	3	
5	2	2.5	2.5	2	3	3	2	
4	2	2.0	2	2	2	2	2	
3	2	1.5	1.5	1	2	2	2	
2	2	1.0	1	1	1	1	1	
1	2	0.5	0.5	0	1	1	0	
計	55	-	27.5	27.5	25	30	30	27

※ 当アプリで確認するには、まずスライダで端数処理の方法と桁数を選択し、計算または数値の入力後に [=] を押します。

備考：「標準」と「拡張」について

この区別は パソコンでの「計算方式」を決める「型」の選択になります。

「標準」では「10進数型・浮動小数点数(Decimal)」を用いています。(一部の関数ボタンでは[Double]を使用)

パソコンソフト用としてあらかじめ用意されている型のなかでは やや速度が遅い面がありますが誤差が少なく信頼性の高いものです。財務計算などにも広く利用されていますので、これを「標準」にしています。

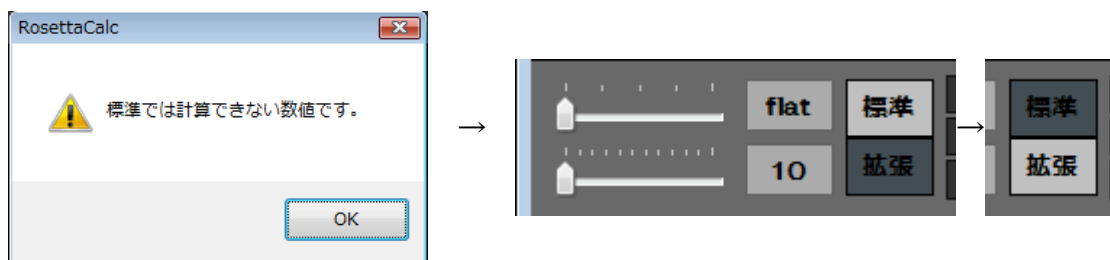
ただ、桁数があまりにも大きい場合や、細かすぎる小数などには不向きなようです。

整数では30桁以上になるとエラーになりますし、小数でも30桁ほどが限界のようです。(32ビットOSの場合)

つまり計算結果としては [無量大数](一般的な表現では 10^{68}) の位までは扱えないことにはなりますが、それでも [億][兆][京][垓]などの大数や、小数では [涅槃寂靜] (10^{24}) など、普通の計算に利用するには十分な範囲といえます。

なお、入力した数値が「標準」で扱える範囲を超えてしまっているときには、計算をおこなってみた段階で(直算電卓では随時)計算できない旨のメッセージが表示されますので、「拡張」へと切り替えるべき基準がわかるようになっています。

計算できるかどうかを簡単にチェックするには、数値の入力後に [=] キーを押します。(直算電卓では入力のみで可) [標準]で計算できない旨のメッセージが表示された場合には、[拡張]に切り替えてお試しください。 ↓



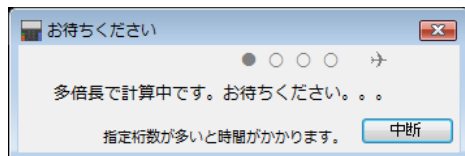
「拡張」では「多倍長整数型(BigInteger)」を用いて、桁数の範囲を広げます。(設定画面「速算」ありの場合)

用意された「型」としては比較的新しいもので、巨大数の計算には向いていますが「整数型」のため、小数部の計算にはそのままの形では使えません。このように利用には一定の工夫を必要とするため、あまり多用される型ではないようです。

当アプリでは、整数部での範囲の上限は特に設けていませんが、数万桁程度なら高速に計算できています。

小数部の計算に関しても、小数点の位置を管理して、多倍長整数型に合うよう、整数に直して計算するなどの工夫や小数部の有効桁数には最大桁数の制限を設ける等で、計算速度をあまり犠牲にすることなく、比較的高速に計算できます。

桁数を広げる場合は、こちらをお使いください。(設定画面の「速算」チェックは入れたまま。小数部は最大10万桁)



「拡張」では、通常ならば計算に推奨される「標準」的な手法とは異なる方法を採用しているため、桁数が多い場合や、複雑な関数などでは、計算に時間がかかることがあります。その場合には図のような画面が表示されます。また必要に応じて、時間のかかっている計算を「中断」することもできます。

ちなみに「拡張」で、設定画面の「速算」チェックが解除された場合には

電子計算機の基本に忠実に作成した、特別な手法を用います。(この方式は計算に時間がかかるため、お薦めできません。)

前述の多倍長整数型が使えない場合のために作成したもので、近年のCPUパワーの向上により実現可能となったものです。数値を文字列で捉えて2進数に直し、加算器なども自作して、「文字列」による「計算」をシミュレートしています。

(当アプリはバージョンアップにより多倍長整数型が使える環境へ移行しており、この方式を用いる必要はありません)

この場合、桁数に制限のある「数値型」は使えないため、型としては「文字型(String)」を使っていることとなります。

(文字型で扱えるのは約20億個のUnicode文字です。桁数も約20億桁となるため、制限がないわけではありません。)

計算効率にはあまり配慮をしていないため、桁数の増加に伴い、計算に時間がかかるようになります。

特に除算時の小数部の計算は複雑で、多倍長整数型ではおこなっている小数部の工夫がない、愚直な方式を採用しています。割り切れない場合のストッパーの必要性和、端数処理の切り上げ時には有効桁数の末尾に1を足す再計算もおこなうため1回の計算にも非常に時間がかかるようになります(小数部の桁数が増えるに応じて累進的に遅くなります)。

(ここでいう「計算」には、四則演算や関数ボタン機能はもちろんですが、表現上の「切替」(変換)でもn進法の表現や進法入力などでは「基数変換」等があるため、それらに含まれる「計算」も指しています。)

(特に小数部が数千桁以上におよぶ計算をおこなう場合は、計算に要する時間が数時間以上におよんだり

「人間が待てる限界」を超えてしまう可能性があります。数日を要しても構わない場合は別ですが

待てなければ最終的に当アプリを強制終了せざるを得なくなることがありますのでご注意ください。)

このため「拡張」での小数部の桁数には段階的な制限を設けており、最大でも10万桁までとしています。

(この制限は多倍長整数型の場合にも引き継いでいますが、実用上も妥当な線だろうと考えています。)

結果として、「拡張」での、設定画面の「速算」チェックの有無による違いは、「計算時間の違い」だけのようです。

したがって「速算」を外した「拡張」につきましては、通常は「計算」には利用されないよう、強くお薦めします。

関数ボタンは、通常の電卓画面で [標準] [拡張] を問わず利用できます。

- 関数ごとに使い方や意味合いが異なります。よく把握してからお使いください。
- 2つの数値を用いる関数では [+][−][×][÷] や [=] と併用します。(後者では表示域右上に演算記号を表示。)
- [直算電卓] や [進法入力] 時には、誤用や誤解を避けるため、関数ボタンが非表示となり、利用できなくなります。
- 一部の関数では、使っている「型」の違いにより、巨大数の大きさや小数の細かさの範囲が異なるものがあります。
- 関数によっては対象数値に制限がある場合があります。(→計算の前や後に注意喚起のメッセージが表示されます。)
- [標準] では結果が巨大数の場合などに **指数表記** となることがあります。(→計算が継続できない場合があります。)
- [拡張] でも極端に大きな数値や極端に細かな小数については、処理能力を超えるために計算できない場合があります。

(1行目: [f1]~[f8])

1. **√** **平方根** 数値Aの平方根(二乗すると元の数値に戻る値。ルート)を求めます。
(使い方: [A] **√**)

例: [2] **√** → [1. 4142135623731] (2の平方根: ルート2: ヒトヒトヨヒトミコロ) [標準]
[1. 41421356237309504880168872420969807856967187537694] [拡張]
[3] **√** → [1. 73205080756888] (3の平方根: ルート3: ヒトミニコレヤ) [標準]
[1. 73205080756887729352744634150587236694280525381038] [拡張]
[5] **√** → [2. 23606797749979] (5の平方根: ルート5: フジサツク(ニ)、オムナク) [標準]
[2. 23606797749978969640917366873127623544061835961152] [拡張]
[256] **√** → [16]
[152399025] **√** → [12, 345]

2. **^2** **二乗** 数値Aを二乗した値(2回掛けた値。自乗、平方)を求めます。
(使い方: [A] **^2**)

例: [16] **^2** → [256] ([16×16]: 16の2乗: 16²)
[256] **^2** → [65, 536] ([256×256]: 256の2乗: 256²)
[12345] **^2** → [152, 399, 025] (12345の2乗: 12345²)
[123456789] **^2** → [15, 241, 578, 750, 190, 521] [拡張]

3. **π** **円周率** 円周率(3. 1415926...。π=パイ)の近似値を求めます。
(使い方: **π**)

例: **π** → [3. 14159265358979] [標準]
[3. 14159265358979323846264338327950288419716939937510] [拡張]

([拡張]の円周率はその都度「**ガウス・ルジャンドルの公式**」(**サラミン・ブレントの公式**)により計算しています。
なおアプリでは設定の変更で小数点以下10万桁まで求めることも可能です(数分ほどかかります。)

利用法: 円の公式 [半径rの円周: $2\pi r$] [半径rの円の面積: πr^2]

[2][×][2][×]**π**[=] → [12. 56637061435916] (半径2の円周または面積)

(扇形の面積や弧の長さを求める場合は、円に対する割合[角度/360度]を掛ければ求められます。)

(数学問題で円周率が[3. 14]など短く決められた場合に求めるのは概算値ですので **π** は使わずに直接入力してください。
近似値では逆に不正解とされることがあります。同じ値の繰り返し入力为了避免には **メモリー機能 [MRC]** が便利です。)

4. **%** **百分率** 元の値に百分率(パーセント)を掛けた値で計算します。
(使い方: 四則演算の [=] の箇所 **%** を押す。)

例: [2000][×][25]**%** → [500] (元の値の25%の値) ← **【基本】**
[2000][+][25]**%** → [2, 500] (元の値に25%の増加)
[2000][−][25]**%** → [1, 500] (元の値に25%の減少)
[2000][÷][25]**%** → [4] (元の値の25%の値で元の値を割った結果)

※ 一般的な電卓の機能とは結果が異なる場合があります。(より実用的となるよう当方で工夫したものです)

5. **1/x** 逆数 1 を元の数値Aで割って、元の数値Aの逆数を求めます。
(使い方: [A] **1/x**)

例: [200] **1/x** → [0.005]
[0.005] **1/x** → [200]

6. **int** 整数化 数値Aの小数部を取り除いた整数値にします。
(使い方: [A] **int**)

例: [1.4142135623731] **int** → [1]
[2.23606797749979] **int** → [2]
[3.14159265358979323846264338327950288419716939937510] **int** → [3]

7. **税抜** 税抜 もとの数値を「税込額」とし、税率を用いて「税抜額」を求めます。
(使い方: [A] **税抜**) (詳細後述)

例: [1080] **税抜** → [1,000] **税抜** → [80] **税抜** → [1,080]
(税率8%の場合)

8. **税込** 税込 もとの数値を「税抜額」とし、税率を用いて「税込額」を求めます。
(使い方: [A] **税込**) (詳細後述)

例: [1000] **税込** → [1,080] **税込** → [80] **税込** → [1,000]
(税率8%の場合)

(2行目: [control]+[f1]~[f8])

1. **n!** 階乗 1 から数値Aまでの整数をすべて掛け合わせた値(積)を求めます。
(使い方: [A] **n!**) ※ 0! = 1 (数学上の規定)。Aが負の数でも結果は正の数

例: [5] **n!** → [120] (1×2×3×4×5)
[6] **n!** → [720] (1×2×3×4×5×6)
[10] **n!** → [3,628,800] (1×2×3×4×5×6×7×8×9×10)
[30] **n!** → [265,252,859,812,191,058,636,308,480,000,000] [拡張]

2. **^n** 冪(累乗) 数値Aどうしの掛け算をB回分繰り返した値(AのB乗: A^B)を求めます。
(使い方: [A] **^n** [B] [=]) (表示域演算記号 ^)

例: [2] **^n** [8] [=] → [256] (2の8乗: 2⁸)
[2] **^n** [10] [=] → [1,024] (2の10乗: 2¹⁰)
[2] **^n** [16] [=] → [65,536] (2の16乗: 2¹⁶)
[10] **^n** [10] [=] → [10,000,000,000] (10の10乗: 10¹⁰)
[10] **^n** [10] **+/-** [=] → [0.0000000001] (10のマイナス10乗: 10⁻¹⁰)
☆ [10] **^n** [10.5] [=] → [31,622,776,601.6838] (10の10.5乗: 10^{10.5})

※ A, Bは負の数も可です。Bに小数を含む場合(☆)は [標準] のみ対応しています(冪)。
([拡張] でのBは整数のみに対応します(累乗))

(以下、2行目の3.~7.の項目は「角度・三角関数」系統でまとめています。)

7. **角度** [角度] / [ラジアン] 切替
 2行目の 3. [sin] ~ 6. [rad]/[deg] の数値が [角度]か[ラジアン]か を選んでおきます。
 (ボタンを押すと色も切り替わります。) **角度** **rad/deg** (計算はおこなわれません)

(角度をラジアン(π)で表現する方法は[弧度法]) **rad** **deg** (6. [rad]/[deg]も切り替わります。)

※ 3. [sin] ~ 6. [rad]/[deg] の数値が角度で [度分秒]表記 (a度b分c秒 / a° b' c'') となっている場合は
 あらかじめ [十進]表記 (小数部を含む[度]のみの表記)に直しておく必要があります。

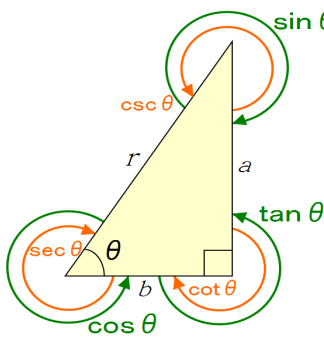
角度の[分][秒]も時間の[分][秒]と同じ「六十進法」(六十分法)が用いられており [1度=60分] [1分=60秒] ですので
 [度]単位に直すには [1分=1/60度=0.0166666度] [1秒=1/60分=1/3600度=0.000277777度] のように計算します。

3. **sin** **三角関数**:サイン(正弦) 直角三角形の内角Aに対する正弦値($\sin \theta$)を求めます。
 (使い方: [A] **sin**)

4. **cos** **三角関数**:コサイン(余弦) 直角三角形の内角Aに対する余弦値($\cos \theta$)を求めます。
 (使い方: [A] **cos**)

5. **tan** **三角関数**:タンジェント(正接) 直角三角形の内角Aに対する正接値($\tan \theta$)を求めます。
 (使い方: [A] **tan**) ($\tan \theta = \sin \theta / \cos \theta$)

直角三角形においては、直角を除いた2つの内角(鋭角)のうち1つの角度 θ (A)が決まれば、3辺の長さの比が決まります
 三角関数は、うち2辺に着目して、決まった $\angle \theta$ に対する2辺のとり方により定まる、2辺の比率をあらわすものです。



直角三角形で、直角と $\angle \theta$ の位置が図のように決まるとき
 $\angle \theta$ に対する2辺の比を示す三角関数は、次のようにあらわされます。

$$[\sin \theta = a/r] \quad [\cos \theta = b/r] \quad [\tan \theta = a/b]$$

※ その他の三角関数: $\sec \theta$, $\csc \theta (= \operatorname{cosec} \theta)$, $\cot \theta$
 (セカント、コセカント、コタンジェント: 正割、余割、余接) については
 当アプリでは $\cos \theta$, $\sin \theta$, $\tan \theta$ (コサイン、サイン、タンジェント) の
 逆数($[1/x]$)をとれば 求めることができます。(辺のとり方による順番に注意)

$$[\csc \theta = r/a] \quad [\sec \theta = r/b] \quad [\cot \theta = b/a]$$

$$[\csc \theta = 1/\sin \theta] \quad [\sec \theta = 1/\cos \theta] \quad [\cot \theta = 1/\tan \theta]$$

例: (**角度** の場合: 60°) (**rad/deg** の場合: $1/3 \pi = 60^\circ$)
 [60] **sin** → [0.86602540378444] [1.0471975511966] **sin** → [0.86602540378444]
 [60] **cos** → [0.49999999999999] [1.0471975511966] **cos** → [0.49999999999999]
 [60] **tan** → [1.73205080756889] [1.0471975511966] **tan** → [1.73205080756889]

6. a **rad** **ラジアン** (**角度** の場合に) 角度としての数値Aを [ラジアン] に直します。
 (使い方: [A] **rad**) (ラジアン(弧度)は通常 π であらわされます。 $\pi=180^\circ$ です。)

例: [60] **rad** → [1.0471975511966] (← $60^\circ = 1/3 \pi$)
 [90] **rad** → [1.5707963267949] (← $90^\circ = 1/2 \pi$)
 [180] **rad** → [3.14159265358979] (← $180^\circ = \pi$)
 [360] **rad** → [6.28318530717959] (← $360^\circ = 2 \pi$)

6. b **deg** **角度** (**rad/deg** の場合に) ラジアンとしての数値Aを [角度] に直します。
 (使い方: [A] **deg**) (1ラジアンは半径と同じ長さの弧を持つ扇形の角度です。)

例: [1.0471975511966] **deg** → [60.0000000000001] (← $1/3 \pi \doteq 60^\circ$)
 [1.5707963267949] **deg** → [90.0000000000002] (← $1/2 \pi \doteq 90^\circ$)
 [3.14159265358979] **deg** → [180] (← $\pi = 180^\circ$)
 [6.28318530717959] **deg** → [360] (← $2 \pi = 360^\circ$)
 [1] **deg** → [57.2957795130823] (← 1ラジアン $\doteq 57.3^\circ$: 正三角形の 60° よりもやや鋭角)

(3行目: [Alt]+[f1]~[f8]) (※以下の1.~4.は正の整数のみの対応となります。)
(1.2.で対象の数値が3つ以上ある場合は、電卓での計算には向いていません。表計算ソフトの利用をお勧めします。)

- GCD** **最大公約数** 2つの数値の最大公約数を求めます。(表示域演算記号 G)
(使い方: [A] **GCD** [B] [=]) (互いに割り切れる数(公約数)のうち最も大きいもの)
例: [30] **GCD** [42] [=] → [6]
- LCM** **最小公倍数** 2つの数値の最小公倍数を求めます。(表示域演算記号 L)
(使い方: [A] **LCM** [B] [=]) (互いに共通する倍数(公倍数)のうち最も小さいもの)
例: [30] **LCM** [42] [=] → [210]

- nPr** **順列** n個からr個を取る際の並べ方(順列)の総数Pを求めます。(表示域演算記号 P)
(使い方: [n] **nPr** [r] [=]) ※条件: [n] ≥ [r] > 0
例: [4] **nPr** [2] [=] → [12] ([abcd] → ab-ba, ac-ca, ad-da, bc-cb, bd-db, cd-dc)
(全体から一定の個数を取り出す場合の[並べ方]。順序が違えばカウントする。)
- nCr** **組合せ** n個からr個の取り出し方(組合せ)の総数Cを求めます。(表示域演算記号 C)
(使い方: [n] **nCr** [r] [=]) ※条件: [n] ≥ [r] > 0
例: [4] **nCr** [2] [=] → [6] ([abcd] → ab, ac, ad, bc, bd, cd)
(全体から一定の個数を取り出す場合の[組合せ方]。並べ方の順序は問わない。)

- log** **対数** Aを底とする数値Bの対数を求めます。(表示域演算記号 l)
(使い方: [A] **log** [B] [=]) ($[a^x = b]$ のとき $[x = \log_a b]$)
例: [2] **log** [8] [=] → [3] (← $2^3 = 8$ により $\log_2 8 = [3]$)
[3] **log** [81] [=] → [4] (← $3^4 = 81$ により $\log_3 81 = [4]$)
[10] **log** [1.22] [=] → [0.0863598306747482] (**常用対数** $\log_{10} B$: 底 A = [10])
※ log関数では[拡張]でも[標準]での計算となります。(A, Bは正の数のみ対応)
- ln** **自然対数** 数値Aの自然対数(ネイピア数を底とする対数)を求めます。
(使い方: [A] **ln**) ($[\ln X = \log_e X]$, eはネイピア数=[2.7182818284...])
例: [2.71828182845905] **ln** → [1] (← ネイピア数[e]の自然対数は[1])
[12.1824939607035] **ln** → [2.5] (← ([e]の2.5乗)の自然対数は[2.5])
- EXP** **指数関数** 数値Aの指数関数(ネイピア数を底とする自然指数関数)を求めます。
(使い方: [A] **EXP**) ($[x = e^x]$, eはネイピア数=[2.7182818284...])
例: [1] **EXP** → [2.71828182845905] (← [1]の指数関数はネイピア数[e])
[2.5] **EXP** → [12.1824939607035] (← [2.5]の指数関数は([e]の2.5乗))
[35] **EXP** → [1.58601345231343E+15] (← ※ **指数表記** 小数点とE+[10の乗数]) [標準]
[1, 586, 013, 452, 313, 430. 728129644625774660125176203950134526154266697022452273408697] [拡張]

- mod** **剰余** 2つの数値で除算(割り算)した[余り]を求めます。(表示域演算記号)
(使い方: [A] **mod** [B] [=]) ※四則演算の[÷]ボタンのすぐ上に配置しています。
例: [14] **mod** [5] [=] → [4] (← $14 \div 5 = 2$ あまり [4])

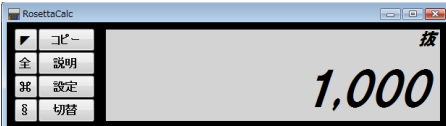
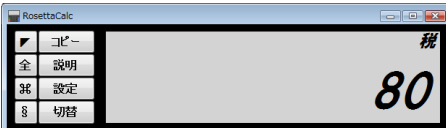
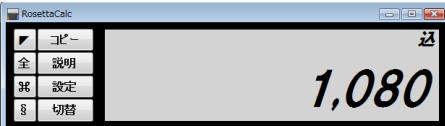
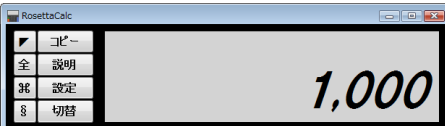
備考：「税抜」「税込」の計算方法



電卓画面に表示中の数値を「金額」と捉えて
それに対する消費税などの「税金」がかかる場合のために、次のような計算をします。

「税抜」… 元の額から税を抜いた「税抜額」（本体価格）を求めます。
（元の額は「税込額」であることを前提にして扱います）

「税込」… 元の額に税を足した「税込額」を求めます。
（元の額は「税抜額」（本体価格）であることを前提にして扱います）

「税抜」の流れ		「税込」の流れ	
(税率8%の場合) (元の額) [1,080] ※税込額と捉える 「税抜」ボタン [抜] (税抜額=本体価格) [1,000]	 ↓  ↓  ↓ 	 ↓  ↓  ↓ 	(税率8%の場合) (元の額) [1,000] ※税抜額(本体価格) と捉える 「税込」ボタン [込] (税込額) [1,080] 「税込」ボタン [税] (税額) [80] 「税込」ボタン (元の額) [1,000] (税抜額=本体価格)

(上の例ではわかりやすい金額での比較にするため、税率を8%にしており、税額が揃うように元の額を変えています。)

以上のように、「税抜」または「税込」のボタンを押す度に

(元の額) → (「税抜額」または「税込額」) → (税額) → (元の額)

の順で、表示が切り替わるようになっています。

税率を用いる計算では、百分率に伴う除算（割り算）によって、小数部が発生することがあります。
小数点以下については、別項「除算」の「小数点以下の有効桁数」の設定によります。

(国ごとの通貨制度で補助通貨を用いる場合（日本では旧補助通貨「銭」など）にも対応可能です)

このとき、特に「税抜」の場合に、端数がどちら側に入るか、ということは、事前に確認しておくべき点となります。

端数については、まず「税額」に対して、別項「端数処理」にて指定された処理をおこないます。

これは、たとえば「消費税の端数は『切り捨て』が良い」とされているなら

ここでは除算時の「端数処理」にて『切り捨て』を選べばいいようにしている、ということです。

この結果、元の額から「端数を切り捨てた消費税」を引いた残りが「税抜額」（本体価格）になります。

(本体価格の側には「消費税の側で切り捨てた端数」がプラスされていることとなります)

このように、正しい計算結果が得られるよう、内部的な計算手順としては、先に税を計算しています。
ただし、ボタンには「税抜」と表示していますので、表示の順序としては「税抜額」から先に表示しています。

(なお「税抜」ではなく「税込」の場合は

「端数を切り捨てた消費税」（端数処理した税額）を、本体価格に足すだけです
——「税込額」を求めるのはそれだけで十分です——ので、特に問題ないものと思われます。)

「税抜」と「税込」の、それぞれの計算方法は、次のようになります。

「税抜」の場合

- 【除数（分母）】 = 【100（%）】 + 【税率（%）】
- 【税額】 = 【元の額（税込額）】 × 【税率（%）】 ÷ 【除数（分母）】
↑
（端数処理）
- 【税抜額】 = 【元の額（税込額）】 - 【税額（端数処理済）】

「税込」の場合

- 【除数（分母）】 = 【100（%）】
- 【税額】 = 【元の額（税抜額）】 × 【税率（%）】 ÷ 【除数（分母）】
↑
（端数処理）
- 【税込額】 = 【元の額（税抜額）】 + 【税額（端数処理済）】

※《参考》 日本の消費税計算のための設定（一般的な用途）

- 税率 ← 10.00（%）（設定画面の項目：2019年10月の改定税率）
- 端数処理 ← 「CUT」（切り捨て）を選択
- 小数点以下の有効桁数 ← 0（桁）（税額で小数第一位を端数として切り捨てて整数化）



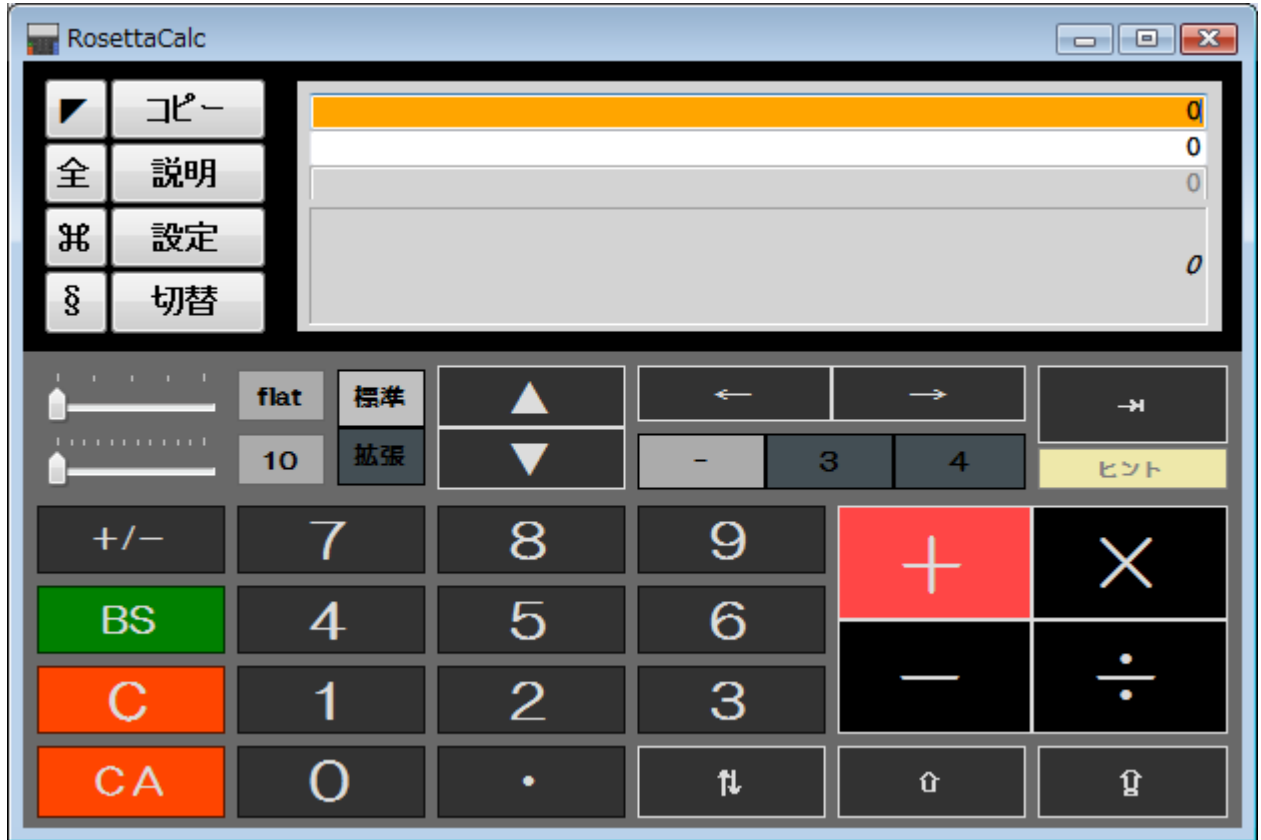
日本の消費税率は、2019年10月に「8%」から「10%」に改定されました。

なお、ここでは日本国内において一般的とされる計算手順をとっていますが、国外では通用しない可能性もあります。国ごとの税制によっては、税の項目が複数に及んだり、補助通貨を持つ場合や端数処理の対象が税額ではないなど、別の方法を探っている場合も考えられます。

端数処理の種類ぐらいなら、切り替えるなどの工夫で対応できる場合もあるかと思われませんが上記の計算手順だけでは残念ながら、世界中の税制すべてには対応できない場合があります。

これに該当する場合は、「税抜」「税込」ボタンは使わずに、電卓の計算機能を使ってその税制に応じた計算方法を別途にとられるよう、お願いいたします。

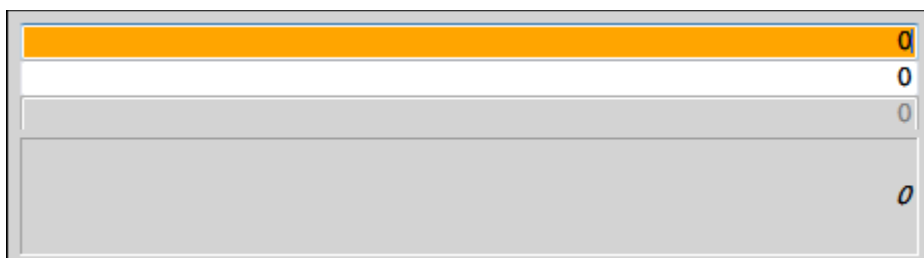
[𠬞] ボタンは、通常の電卓画面と、当アプリ独自の「直算電卓」とを切り替えます。



「直算電卓」とは、あらかじめ選んでおいた加減乗除 [+] [-] [×] [÷] により四則演算をおこなう2つの数値の、片方は固定したまま、もう片方の数値を変化させて常にいずれかの計算をおこない、結果に基づく数値表現も常におこなわれるようにしたものです。

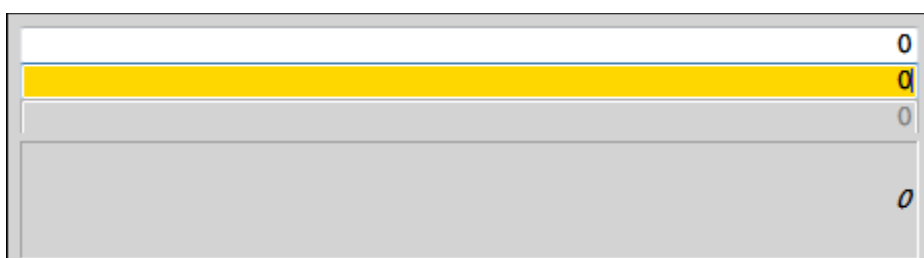
当アプリの多彩な数値表現、および「多桁計算」にも対応できる高度な計算機能はそのままに小学校低学年でも使える「かずあそび」にも、大人の研究者の応用的な要求などにも対応可能なシンプルな操作性を備えたものになっています。

通常電卓での表示域にあたる右上の部分は4段に分かれており、1段目と2段目が数値の入力欄、3段目と4段目が計算結果の表示欄となっています。(1～3段目はアラビア数字のみ、4段目は数値表現です。)



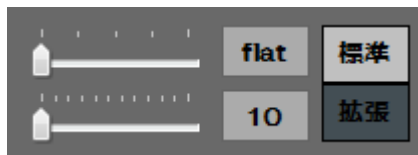
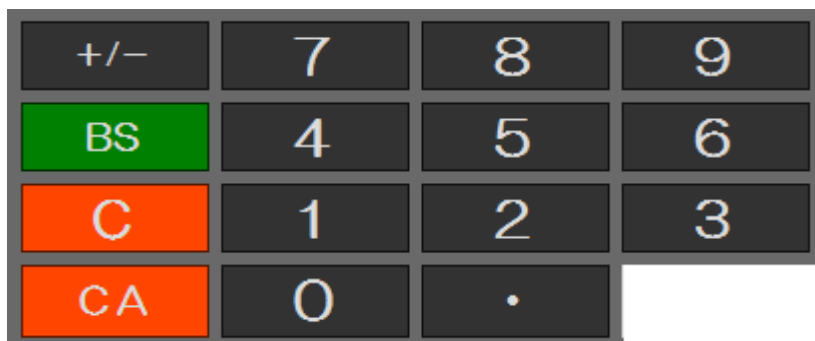
(文字サイズは通常電卓と同様に調節可)

数値入力欄にはキーボードやテンキー等からの直接入力も可能で、入力用のカーソルが点滅します。現在入力対象の欄には色がつき、2段目を選択すると色も変わるので、わかりやすくなっています。



(2段目を選択したところ。)

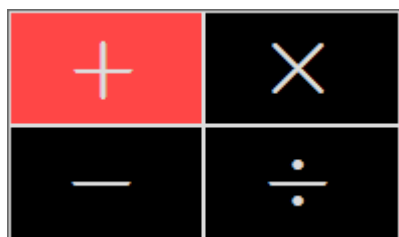
基本の操作系 および 端数処理や[多桁計算]にも対応する操作部も、通常の電卓画面とまったく同じです。



[切替] ボタンによる数値表現は、入力/表示域の4段目に表示されます。
 [+/-] (入力数値の正負の反転)の操作は、キーボードの [-]キー でもできます。
 (進法入力をおこなう場合は 画面左上の操作部が3通りになるのも同じです(後述)。)
 (入力欄の形式が異なるため消去削除系 [BS][C][CA] の操作感は異なります(後述)。)
 (端数処理や有効桁数、[標準]/[拡張]の操作部は、通常電卓の状況を引き継ぎます。)

直算電卓では、選択した計算方法により、入力の進行中にも常に計算結果を表示するため、 [=] キーはありません。
 このほか、通常の電卓画面とは異なる「直算電卓」独自の操作項目には、次のものがあります。

四則演算 (加減乗除 [+][-][×][÷])



2つの数値のあいだで常におこなう「四則演算」を選んでおきます。
 (数値入力の進行中にも計算結果が表示され、演算子はいつでも切り替え可能です。)

右下部分



2つの数値入力欄の内容を、相互に入れ替えます。



四則演算の結果の数値 (3段目) を、2段目の入力欄へ移します。(↑)

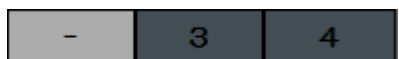


四則演算の結果の数値 (3段目) を、1段目の入力欄へ移します。(↑ : 1つ上)

中段右側部分

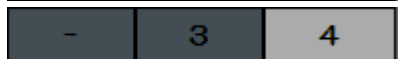


選択中の数値入力欄の、カーソル位置を左右に動かします。

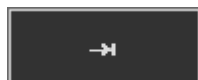


四則演算の結果の数値 (3段目) の、桁区切り表記を選択できます。
 (桁区切り記号はカンマのみです。)

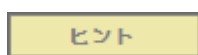
([-] : 桁区切りなし、[3] : 3桁区切り、[4] : 4桁区切り)



選択中の数値入力欄の数値を、1ずつ増減させます。
 ([alt]キーを押しながらかすと、小数部があれば小数部の末尾桁の数値を1ずつ増減させます。)
 ([shift]キーを押しながらかすと、文字の大きさを拡大縮小します。)



2つの数値入力欄の選択を、交互に移動します。
 (キーボードの[TAB]キーと同様。[Enter]や[Return]、上下矢印キー[↑][↓]でも可。)



ツールチップヒントの 表示/非表示 を切り替えます。(通常電卓と同様)

ある数値に対して連続する数値による加減乗除をおこなう場合

加算 [+] 最初は1段目に [5] と入力して2段目へ移動、あとは [▲] ボタン (1 増加) を繰り返す

5	5	5	5	5	5	5	5	5	5	5	5	5	5
0	1	2	3	4	5	6	7	8	9	10	11	12	13
5	6	7	8	9	10	11	12	13	14	15	16	17	18
5	6	7	8	9	10	11	12	13	14	15	16	17	18

減算 [-] 最初は1段目に [5] と入力して2段目へ移動、あとは [▲] ボタンを繰り返す

5	5	5	5	5	5	5	5	5	5	5	5	5	5
0	1	2	3	4	5	6	7	8	9	10	11	12	13
5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8
5	4	3	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8

乗算 [×] 最初は1段目に [5] と入力して2段目へ移動、あとは [▲] ボタンを繰り返す

5	5	5	5	5	5	5	5	5	5	5	5	5	5
0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	5	10	15	20	25	30	35	40	45	50	55	60	65
0	5	10	15	20	25	30	35	40	45	50	55	60	65

除算 [÷] 最初は1段目に [5] と入力して2段目へ移動、あとは [▲] ボタンを繰り返す

5	5	5	5	5	5	5	5	5	5	5	5	5	5
0	1	2	3	4	5	6	7	8	9	10	11	12	13
	5	2.5	1.666666666666666666666666666667	1.25	1	0.83333333333333333333333333333333	0.7142857142857142857142857143	0.625	0.555555555555555555555555555556	0.5	0.454545454545454545454545454545	0.407407407407407407407407407407	0.370370370370370370370370370370
5	2.5	1.666666666666666666666666666667	1.25	1	0.83333333333333333333333333333333	0.7142857142857142857142857143	0.625	0.555555555555555555555555555556	0.5	0.454545454545454545454545454545	0.407407407407407407407407407407	0.370370370370370370370370370370	

ある数値をもとに同じ数値を使った加減乗除を繰り返す場合

加算 [+] 最初は2段目に [5] と入力、あとは [↑] ボタン (結果を1段目へ) を繰り返す

0	5	10	15	20	25	30	35	40	45	50	55	60	65
5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	10	15	20	25	30	35	40	45	50	55	60	65	70
5	10	15	20	25	30	35	40	45	50	55	60	65	70

(通常電卓での $[0 + 5 + 5 + 5 + 5 + 5 + 5 \dots]$ の操作と同じ計算をボタンだけで実現)

減算 [-] 最初は2段目に [5] と入力、あとは [↑] ボタンを繰り返す

0	-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60	-65
5	5	5	5	5	5	5	5	5	5	5	5	5	5
-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60	-65	-70
-5	-10	-15	-20	-25	-30	-35	-40	-45	-50	-55	-60	-65	-70

(通常電卓での $[0 - 5 - 5 - 5 - 5 - 5 - 5 \dots]$ の操作と同じ計算をボタンだけで実現)

乗算 [×] 最初は1段目に [1]、2段目に [5] と入力、あとは [↑] ボタンを繰り返す

1	5	25	125	625	3125	15625	78125	390625	1953125
5	5	5	5	5	5	5	5	5	5
5	25	125	625	3125	15625	78125	390625	1953125	9765625
5	25	125	625	3,125	15,625	78,125	390,625	1,953,125	9,765,625

(通常電卓での $[1 \times 5 \times 5 \times 5 \times 5 \times 5 \times 5 \dots]$ の操作と同じ計算をボタンだけで実現)

除算 [÷] 最初は1段目に [1]、2段目に [5] と入力、あとは [↑] ボタンを繰り返す

1	0.2	0.04	0.008	0.0016	0.00032	0.000064	0.0000128	0.00000256
5	5.0	5.00	5.000	5.0000	5.00000	5.000000	5.0000000	5.00000000
0.2	0.04	0.008	0.0016	0.00032	0.000064	0.0000128	0.00000256	0.000000512
0.2	0.04	0.008	0.0016	0.00032	0.000064	0.0000128	0.00000256	0.000000512

(通常電卓での $[1 \div 5 \div 5 \div 5 \div 5 \div 5 \div 5 \dots]$ の操作と同じ計算をボタンだけで実現)

4段目の数値表現には、設定の初期値で「入力用」に選択された「数字1」の表現が適用されています。片方に小数部が入力されると、もう片方の小数部の桁数を自動調整して、小数点の位置を合わせます。

何度も同じ数値を入力する手間を省くことができるため、間違いも減らすことができます。

上の例では簡単な数値 (1桁の整数 [5]) にしていますが、数値が複数桁などの場合はさらに有効です。

次の例は1段目に [12345]、2段目に [67890] と入力して、四則演算の加減乗除を切り替えてみてください。

加算 [+]

12345
67890
80235
80,235

+	×
-	÷

減算 [-]

12345
67890
-55545
-55,545

+	×
-	÷

乗算 [×]

12345
67890
838102050
838,102,050

+	×
-	÷

除算 [÷]

12345
67890
0.1818382677861246133451171012
0.1818382677861246133451171012

+	×
-	÷

12345
67890
0.18183826778612461334511710119310649580203269995581
0.18183826778612461334511710119310649580203269995581

(多桁計算の [拡張] を選んでから除算に切り替えた場合。小数部は初期値の50桁)

3段目は計算結果の単純な表示域ですが、広がっている4段目は、3段目の計算結果に対応して「通常電卓」と同様に [切替] ボタンで表現の切り替えができる、数値表現の表示域となっています。

(例示用に文字を大きめにしています。)

(楔形数字は[六十進法] ($12345 = 3 \times 60^2 + 25 \times 60 + 45$) です。)

3段目の表示に「桁区切り」を入れると、4段目の数値表現での、表現上の区切り位置との関係がわかりやすくできる場合があります。(3段目での桁区切り記号はコンマのみです)

3桁区切り (「記数法2」ページ・英語の千進法)

4桁区切り (「漢字文化圏」ページ・日本語の万進法)

3桁区切り (「漢字文化圏」ページ・ベトナム語の千進法)

備考：消去削除の操作について

直算電卓には 上下2段の入力欄があり、選んだ入力欄のカーソル位置に直接入力していく形式のため特にキーボードを使つての [BackSpace(BS)] キー、[Delete] キー など、消去や削除系の操作については「メモ帳」などでの操作に似ており、当アプリでの通常の電卓画面とは 操作感が異なる部分があります。

キーボード

- [BackSpace(BS)] キー カーソル（カレット）位置の「左側」を、1文字ずつ削除します。 BS
 （[←(左向きの)delete] キー も可）
- [Delete] キー カーソル（カレット）位置の「右側」を、1文字ずつ削除します。

電卓画面（直算電卓）

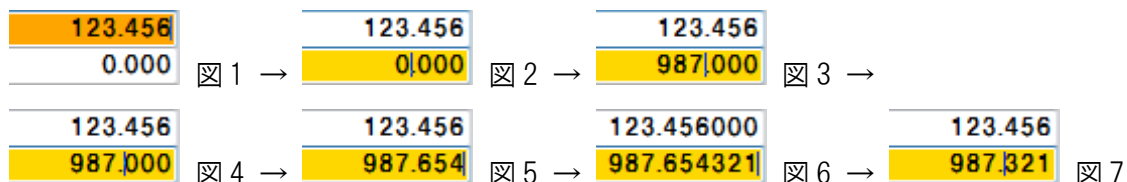
- BS [BS] カーソル（カレット）位置の「左側」を、1文字ずつ削除します。
 （[BackSpace(BS)] キー、[←(左向きの)delete] キー も可）
- C [C] 選択中の入力欄をゼロにします。（[esc(Escape)] キー、[Clear] キー も可）
 （[shift/ctrl]+[Delete] キー、[shift/ctrl]+[BackSpace(BS)] キー も可）
- CA [CA] 上下2段の入力欄を、ともにゼロにします。
 （[alt]+[Delete] キー、[alt]+[BackSpace(BS)] キー も可）

備考：小数部を含む場合の入力

直算電卓では入力域が2段あるため、小数部がある場合には、入力していく際にわかりやすくなるよう小数部の桁数を自動的に上下で揃えて、小数点の位置が揃うようにしています。

これは小数部で片方の末尾が増える度に上下が連動して（当面はゼロが）増えていくものです。

たとえば1段目で普通に [123.456] と入力すれば、カーソル位置(|)は右端にきており2段目は自動的に末尾ゼロの小数部が追加されて [0.000] となります。[図1]



次に、クリック以外の方法（キーボードの上下矢印キーや電卓画面にもある TAB キー等）で2段目の入力に移ると入力域を切り替えた当初にはカーソル位置が小数点の左側（整数部の末尾）へと位置付けられます。[図2]

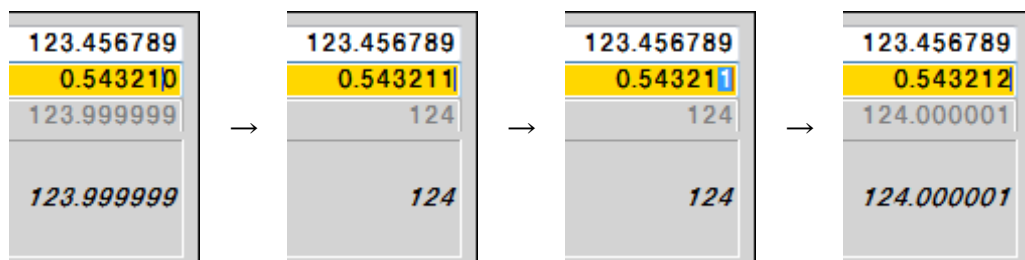
ここで整数値を入力すればそのまま整数値が入りますし（[図3]）、小数部も続けて入力できます（図4-6）。

（整数部と小数部は「小数点(.)」で分けられますので、[図4]のように小数部の入力を始められるようにするには普通に「小数点を入力」してもいいですし「既に表示されている小数点の右側へカーソルを移動」でも構いません。）

（[図6]では小数部を6桁入力したので、1段目でもゼロが追加されて6桁に伸びています。

[図7]では途中の[654]部分を削除して小数部が3桁に減ったため、1段目の末尾の余分なゼロも消えています。）

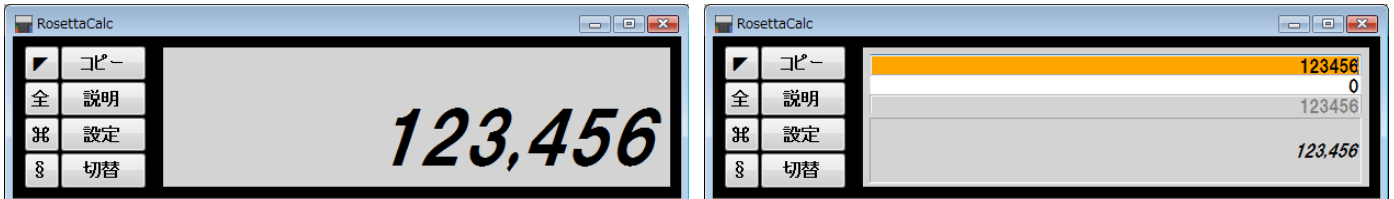
ただし、入力域では小数部の桁数が長くても、計算結果では小数部の桁数が短くなる場合などがあるためそのような場合などには、入力域と計算結果のあいだで、小数点の位置は揃わなくなります。



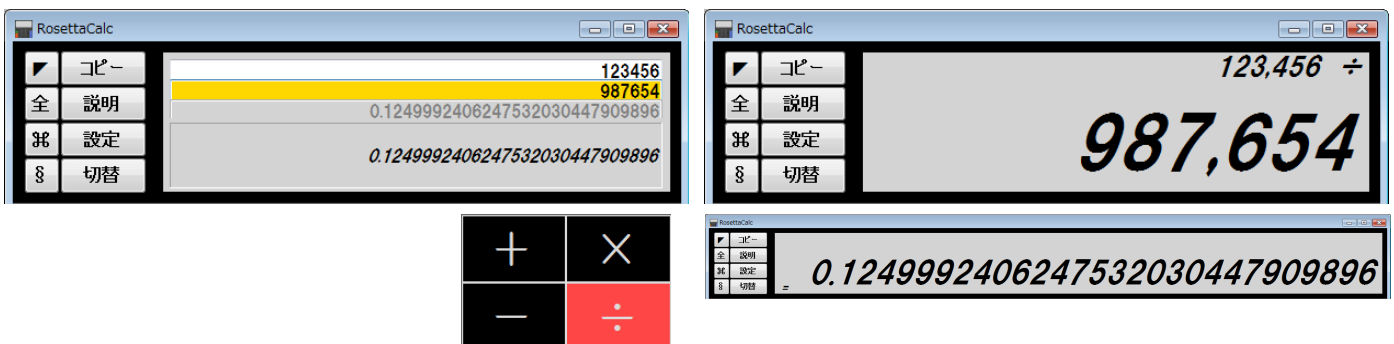
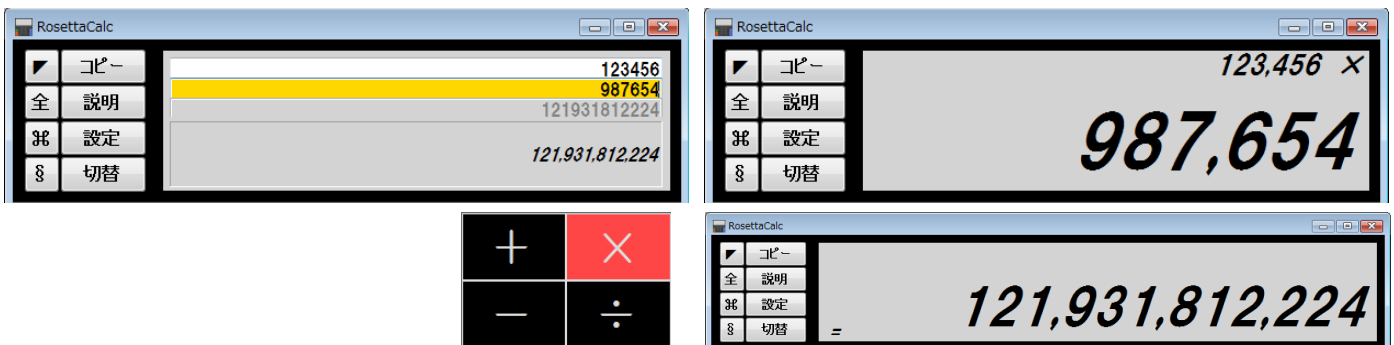
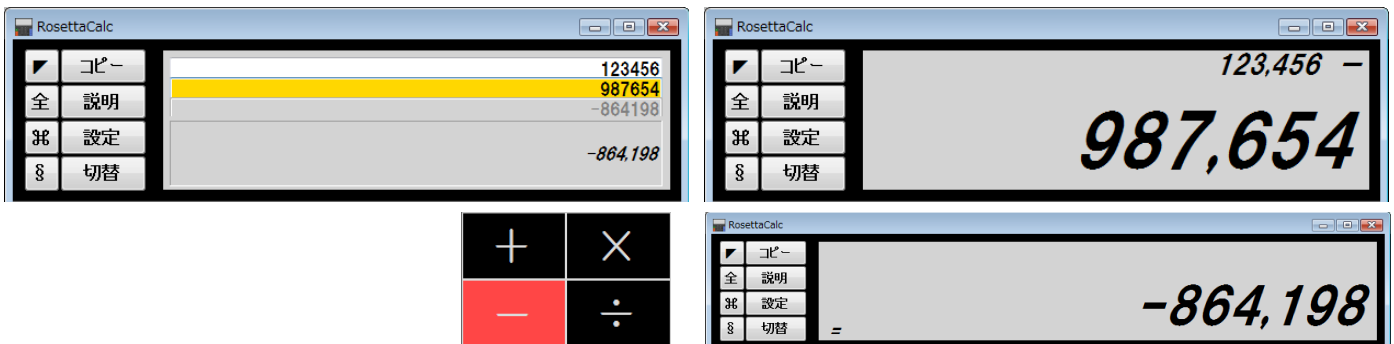
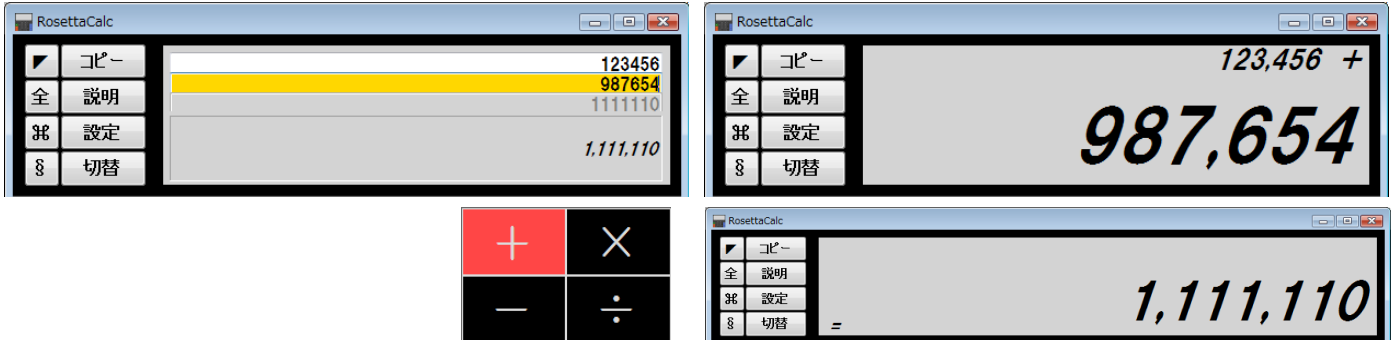
（この例では2段目の小数部の末尾6桁目に[1]が入力されたので、加算の計算結果では小数部が一旦解消されています。入力域とは不揃いですが、2段目の末尾を選択して[2]にすると、計算結果の小数部の桁数も元の6桁に戻っています。）

備考：通常の電卓画面と「直算電卓」との切り替え

入力値がひとつだけの場合は、「直算電卓」の上段側と、相互に切り替わります。



入力値が2つの場合は、「直算電卓」ではつねに計算結果が表示されていますが、通常の電卓画面では演算子が途中にあって「今まさに計算しようとする直前の状態」を示し、 [=] を押せば計算されます。



設定画面で「進法入力」にチェックを入れると、2・8・16進数による入力ができるようになります。

進法入力 (チェックあり・「§」ボタンによる遷移は3通り)



通常入力時(左上)と 進法入力時の電卓画面



各「進法入力」を選んだ際に、使用可能な数字キーは、次のように切り替わります。

- 十進法 [0][1][2][3][4][5][6][7][8][9]
- 二進法 [0]と[1]のみ
- 八進法 [0][1][2][3][4][5][6][7]
- 十六進法 (大文字) [0][1][2][3][4][5][6][7][8][9][A][B][C][D][E][F]
- 十六進法 (小文字) [0][1][2][3][4][5][6][7][8][9][a][b][c][d][e][f]

(関数ボタンや端数処理スライダは、誤用や誤解を避けるため、[十進法]の場合のみ、利用可能にしています。)

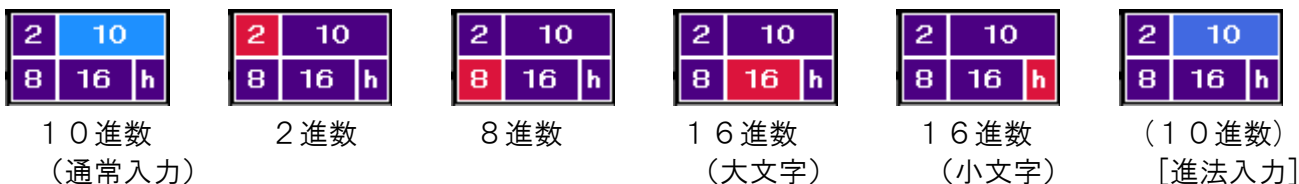
通常の10進数入力(通常入力)では、これまでどおり「入力用」に選んだ数値表現がおこなわれます。

「進法入力」を選んだ場合は、「入力用」に選んだ表現ではなく、設定画面「進法」ページ下段側の「2・8・10・16進数(補数対応・進法入力対応)」に設定済みの数値表現が基本となります。

(10進数で特に「進法入力」を使いたい場合は、左側のチェックボックスにチェックを入れれば利用可能です。)



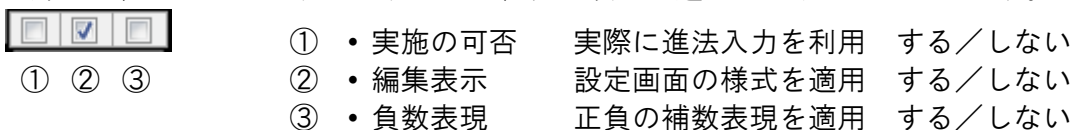
「進法入力」では、2・8・10・16の基数を選択（16進数では大文字小文字も選択可）して選んだ進法に基づいた数字を使つての入力ができます。



左端にある [▲][▼] ボタンでは、入力中の数値を1ずつ増減できます。（整数値）

- [alt]キーを押しながら押すと、小数部の末尾桁で1ずつ増減します。（小数部が無ければ整数部）
- [shift]キーを押しながら押すと、文字の大きさを拡大縮小します。

これら以外にも、3つのチェックボックスで、次の項目が選べるようになっています。



このうち ① については、基数を10から2・8・16へ変更すると、自動的に「する」になります。その状況下で「しない」に変更すると、選択中の基数に関係なく「入力用」として選択された十進法での通常入力に戻ります。これは選んでおいた基数を変えずに「十進法の通常入力」と「選んだ基数の進法入力」とを切り替えて使うのに便利です。

そして ② 「編集表示」は、設定画面「進法」ページ下段の「2・8・10・16進数（補数対応）」の次の項目で設定した様式を、まとめて適用するかどうかの選択になります。

- 桁区切り（しない／する）、桁区切り桁数（スライダ）
- 変換結果に追加する定型的表現（基数の明確化など）

また ③ 「負数表現」については、進法入力ではシンプルにチェックボックスで選べるよう

- 一般（チェックなし） … 「一般(マイナス符号)」
- 補数（チェックあり） … 「基数の補数」

の2択にして 設定画面の「負数表現」の2項と対応しており、チェックありの「補数」を選んだ場合は設定画面で指定された整数部桁数により、整数部の桁数一定化が必ずおこなわれるようにしています。

- 整数部桁数（数値指定）



前述のとおり、上図「進法」ページで設定した表現様式が、「進法入力」（2・8・16進数での入力）時に反映されます。（負数表現は、進法入力においては「一般」または「補数」（基数の補数）の2種類のみで、電卓画面にて切り替えます。）

「2進数」の場合の状況変化（8進数・16進数も基本的に同様）。設定：桁区切り4桁・整数部16桁。

	<p>a1. ① 進法入力 [しない] → 十進法【通常入力】の状態。「入力用」として選んだ「数字1」に設定したフォントでの表示/入力。桁区切りのコンマは「数字1」の指定による。</p>
	<p>a2. 2進数の [2] を選択。 ① 進法入力 [する] (自動) → 基数変換「二進法」の進法入力。 ② 編集表示 [する] (既定)</p>
	<p>a3. 以降も「二進法」の進法入力。 ① 進法入力 [する] ② 編集表示 [しない] → 数値のみの表示となる。</p>
	<p>(a2-b.) ① 進法入力 [する] ② 編集表示 [する] ③ 負数表現 [補数]</p>
	<p>(a3-b.) ① 進法入力 [する] ② 編集表示 [しない] ③ 負数表現 [補数]</p>
	<p>a4. ① 進法入力 [しない] を選択。 → 2進数の [2] の選択は残したまま 基数変換され【通常入力】の状態に戻る。 (「二進法入力」は終了し「十進法入力」が再開。)</p>

「10進数」の場合の状況変化。設定：桁区切り3桁・整数部15桁。

	<p>b1. ① 進法入力 [しない] → 十進法【通常入力】の状態。「入力用」として選んだ「数字1」に設定したフォントでの表示/入力。桁区切りのコンマは「数字1」の指定による。</p>
	<p>b2. ① 進法入力 [する] を選択。→ 「十進法」の進法入力。 ② 編集表示 [する] (既定) (基数かわらず) 「10進数(補数対応)」用に設定されたフォントでの表示。 編集表示による桁区切りと基数明記。</p>
	<p>b3. 以降も「十進法」の進法入力。 ① 進法入力 [する] ② 編集表示 [しない] → 数値のみの表示となる。</p>
	<p>(b2-b.) ① 進法入力 [する] ② 編集表示 [する] ③ 負数表現 [補数]</p>
	<p>(b3-b.) ① 進法入力 [する] ② 編集表示 [しない] ③ 負数表現 [補数]</p>

備考：「補数」について

$a + b = c$ の関係において

a に対して足した結果の c が 1 桁増える最小の数 b を「補数」といいます。

たとえば、通常の【10進数】（十進法）においては

「0～9」の10個の数字を用いて1桁分の数値をあらわし

「9」の値を超えると繰り上がるので「10」となり1桁増えます。

というルールがあるので、次のような関係が成り立ちます。

$1 + 9 = 10$ なので「9は1に対する補数」であり「1は9に対する補数」である。

$2 + 8 = 10$ なので「8は2に対する補数」であり「2は8に対する補数」である。

$3 + 7 = 10$ なので「7は3に対する補数」であり「3は7に対する補数」である。

$4 + 6 = 10$ なので「6は4に対する補数」であり「4は6に対する補数」である。

$5 + 5 = 10$ なので「5は5に対する補数」である。

簡単ですね。2桁以上の場合も同様です。

$01 + 99 = 100$ なので「99は01に対する補数」であり「01は99に対する補数」である。

$02 + 98 = 100$ なので「98は02に対する補数」であり「02は98に対する補数」である。

⋮

$10 + 90 = 100$ なので「90は10に対する補数」であり「10は90に対する補数」である。

$11 + 89 = 100$ なので「89は11に対する補数」であり「11は89に対する補数」である。

⋮

$50 + 50 = 100$ なので「50は50に対する補数」である。

⋮

$0001 + 9999 = 10000$ なので「9999は0001に対する補数」であり「0001は9999に対する補数」である。

なお、「1」は「9に対する補数」であって「9999に対する補数」ではないことに注意してください。

「9999に対する補数」は「0001」です。つまり前提として「桁数を決めておく」こととなります。

また、足した結果が「1桁増える」部分は考えないようにします（桁あふれ）。これらの特性を活かして負の数の表現として「補数」を使えば、「引き算の結果」を「足し算によって求める」ことができます。

$$[26] - [11] \rightarrow [26] + [-11] \rightarrow [26] + [89] = [115] \rightarrow [26] - [11] = [15]$$

この例では2桁の数値で計算しましたが、もっと大きな数値にも対応できるようにするにはあらかじめ十分な余裕をみて、桁数を増やしておく必要があります。

ここでは例として4桁に揃えたとすると、前出の定義は次のように書き直すことができます。

$0001 + 9999 = 10000$ なので「9999は0001に対する補数」であり「0001は9999に対する補数」である。

$0002 + 9998 = 10000$ なので「9998は0002に対する補数」であり「0002は9998に対する補数」である。

⋮

$0011 + 9989 = 10000$ なので「9989は0011に対する補数」であり「0011は9989に対する補数」である。

これにより、「負の数」の表現に4桁の「補数」を使った計算の部分は、次のようにも書けます。

$$[0026] + [9989] = [10015] \quad ([26] + [-11] = [26] - [11] = [15])$$

以上は【10進数】のため、補数の関係は「 a は b に対する【10の補数】である」のようにもいいます。

10進数の[10]のことは「基数」と呼びますから、もっと正確には

「 a は b に対する10進法の【基数の補数】である」という言い方をすることもあります。

わざわざこんな言い方をするのは、ほかに【減基数の補数】もあるので、これとの区別と、それ以外にも10以外の n を基数にとる「 n 進法」にも、同じように「補数」を求めることができるためです。

「n進法」でよく使われる基数 [2][8][16] を例にとると、それぞれのルールは次のようでした。

【2進数】（二進法）では

「0～1」の2個の数字を用いて1桁分の数値をあらわし

「1」の値を超えると繰り上がるので「10」（イチゼロ）となり1桁増えます。

【8進数】（八進法）では

「0～7」の8個の数字を用いて1桁分の数値をあらわし

「7」の値を超えると繰り上がるので「10」（イチゼロ）となり1桁増えます。

【16進数】（十六進法）では

「0～9 および A～F」の16個の数字と英字アルファベットを用いて1桁分の数値をあらわし

「F」（=15）の値を超えると繰り上がるので「10」（イチゼロ）となり1桁増えます。

なお、2進数においては「[2の補数](#)」や「[1の補数](#)」という言い方がよく使われます。これらはすなわち「2の補数」＝【基数の補数】 「1の補数」＝【減基数の補数】 をあらわしています。

「1の補数」＝【減基数の補数】は、もとの数値の各桁を「ビット反転」させて簡単に得られるものです。
「2の補数」＝【基数の補数】は、減基数の補数に1を足せば求めることができ、結果は前頁と同じです。

2進数には [0]と[1] しかなく、[1]+[1]=[10]（イチゼロ）なので「1は1に対する補数」が良いですがこれだけでは補数で1以外の数値が表現できません。そこで一旦【減基数の補数】の考え方を使います。

以下は10進数の [26]と[11] を、各基数法と 減基数の補数、基数の補数 であらわしてみたものです。
(基数ごとで表現に必要な桁数が異なるため、以下では8桁(桁区切り4桁)に揃えています。)

	[26] n進法	減基数の補数	基数の補数	[11] n進法	減基数の補数	基数の補数
【2進数】	0001 1010	1110 0101	1110 0110	0000 1011	1111 0100	1111 0101
【8進数】	0000 0032	7777 7745	7777 7746	0000 0013	7777 7764	7777 7765
【10進数】	0000 0026	9999 9973	9999 9974	0000 0011	9999 9988	9999 9989
【16進数】	0000 001A	FFFF FFE5	FFFF FFE6	0000 000B	FFFF FFF4	FFFF FFF5

「補数」は、パソコンや電卓などのデジタル機器の内部（2進数）で数値を扱う際に、負の数の場合には各桁を「ビット反転」させた「1の補数」＝【減基数の補数】をもとに「2の補数」＝【基数の補数】を得て、シンプルな「加算」によって「減算結果を得る」などの方法に用いられています。

(「基数の補数」「減基数の補数」「絶対値表現」(符号ビット)につきまして、詳しくは別稿「2・8・10・16進数(補数対応・進法入力対応)」の説明をご参照ください。)

● 補数を使った計算が意味を持たない場合

ここまで見てきた特性により、補数は「加算」(足し算)および「乗算」(掛け算)については有効ですが「減算」(引き算)や「除算」(割り算)に「補数で表現した負の数」を使っても、正しい結果は得られず無意味となります。

10進数で、前出の [26]と[11] を4桁の補数として実際に計算してみると、次のようになります。
(左側は桁あふれを施した完全な補数表現として記載。右側の括弧内は補数を使わない書き方)

加算	[0026] + [9989] = [0015] → [15]	◎ ([26] + [-11] = [26] - [11] = [15])	正解
減算	[0026] - [9989] = [9963] → [-37]	△ ([26] - [-11] = [26] + [11] = [37])	再変換 正負逆転
乗算	[0026] × [9989] = [9714] → [-286]	○ ([26] × [-11] = [-286])	正解(要再変換)
除算	[0026] ÷ [9989] = [0000.0026028631...] ×	([26] ÷ [-11] = [-2.363636363...])	不一致(不正解)

(ちなみに正の数の場合は[0026] ÷ [0011] = [0002.363636363...]などのようになるので普通に計算可能です。)

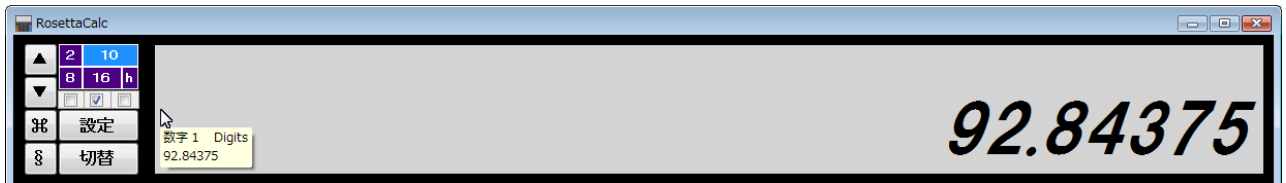
このため、補数は一般的には「減算のための利便性」を活かすべく「加算」のみで使われます。

進法入力・基数変換例

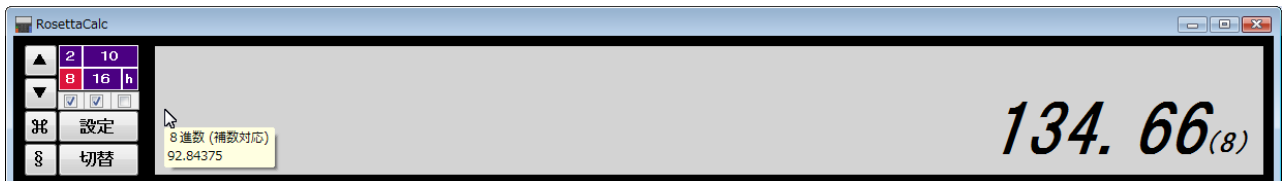
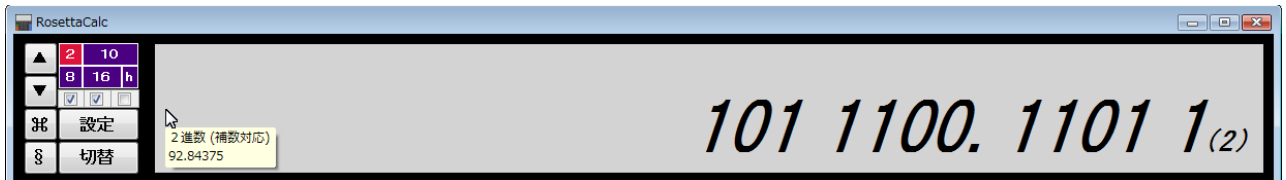
(それぞれ 上から 2・8・10・16進数での進法入力へ切替)
 (桁区切り 4 桁・整数部は 16 桁。10進数の進法入力では別途チェックを入れて表示)



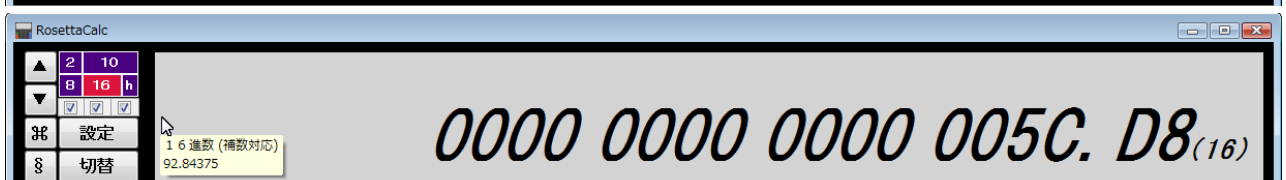
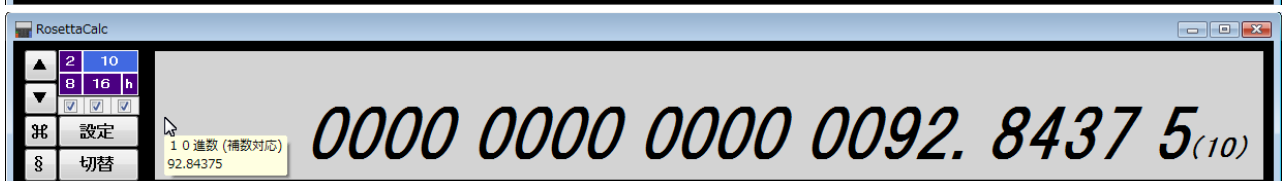
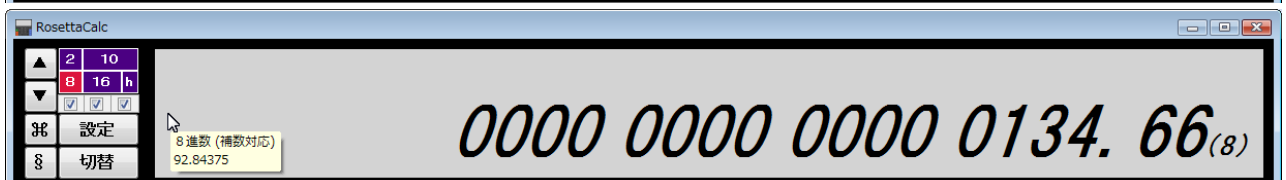
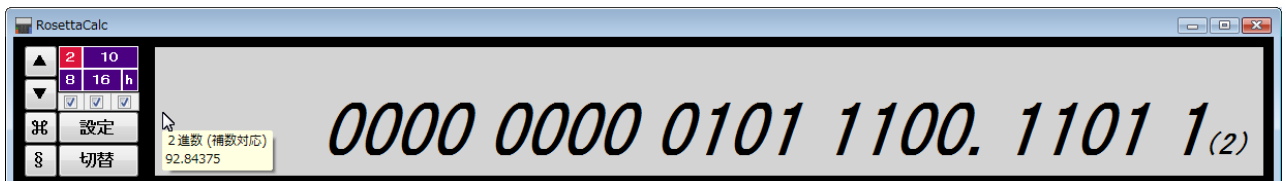
正の数 (10進数で [92.84375]、進法入力 [しない] = 通常入力)



① 進法入力 [する] ・ ② 編集表示 [する] ・ ③ 負数表現 [一般]



① 進法入力 [する] ・ ② 編集表示 [する] ・ ③ 負数表現 [補数]



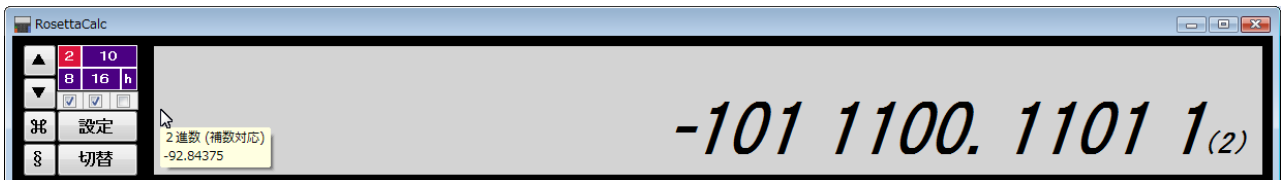
※ 基数変換について詳しくは、別稿「2・8・10・16進数（補数対応・進法入力対応）」をご参照ください。



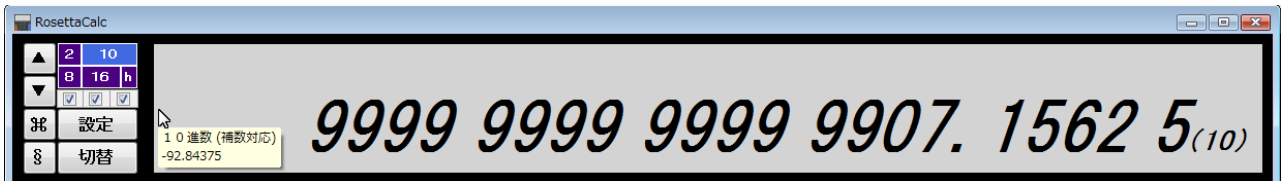
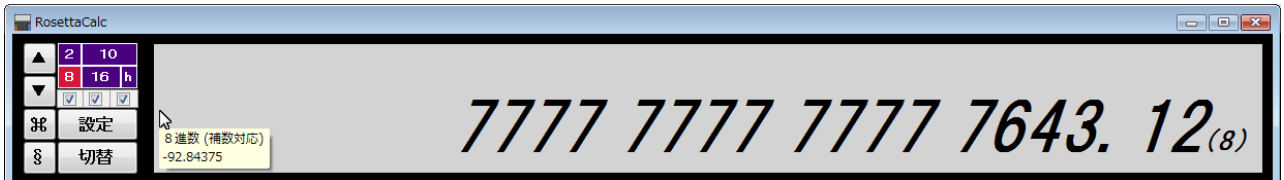
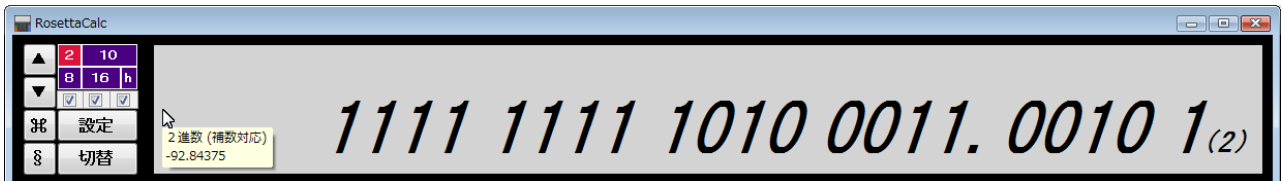
負の数（10進数で [-92.84375]、進法入力 [しない] =通常入力）



① 進法入力 [する] ・ ② 編集表示 [する] ・ ③ 負数表現 [一般]



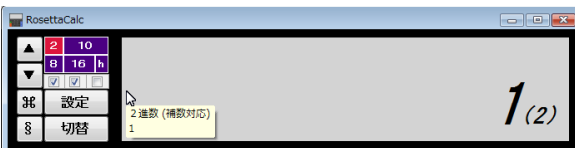
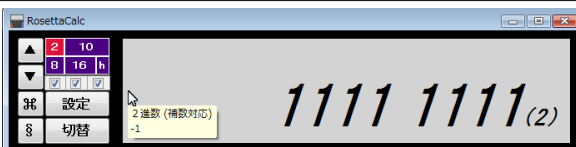
① 進法入力 [する] ・ ② 編集表示 [する] ・ ③ 負数表現 [補数]



備考：「進法入力」時の注意事項

●「補数表現」と「正負の判定」

進法入力時に負の数を表示するのに「補数表現」を使うと、「整数部桁数一定化」がおこなわれ正の場合は [0]、負の場合は選択した基数に応じた「符号ビット」で、先頭部分が埋まります。

負数表現「一般」・桁区切り4桁	負数表現「補数」(基数の補数)・桁区切り4桁・整数部8桁
[0] 	[0] 
[1] 	[1] 
[-1] 	[-1] 

(例示用に整数部桁数を減らして8桁に設定しています。なお設定の上限は1万桁です。)

これはもちろん「正負の判定」に使われるものなので、なるべく「先頭の1桁」だけは残すようにして変更したくないところですが、直接入力していく「進法入力」においては、簡単に変更できてしまいます。

通常、電卓画面の [切替] ボタンから表示を切り替えできる「n進法」(基数法)の「表現」では元の数値は必ず「10進数」であり、それをもとに「基数変換」によって表現されるのですが「進法入力」ではその基数法で直接「入力」していくものなので、そのような問題が起こり得ます。

これを防止するには事前の設定で「整数部桁数」に十分な余裕を持たせておけば良いのですが、それでも設定した桁数を超える数値が入力されてしまった場合、先頭の1桁は必ず「符号ビット」と捉えるため先頭の1桁に「負の数をあらわす符号ビット」が入ってしまうと「正負の逆転」が発生して一瞬にして予期しない数値になってしまうという現象が起こります。(これは間違いではありません。)

[1111111]と入力([1]を7個)。値は10進数で[127]。 	[1111111]と入力([1]を7個)。値は10進数で[127]。 
↓ 8個めの[1]を入力。10進値は[255]。 	↓ 8個めの[1]を入力。10進値は[-1]となる。【正負逆転】 
↓ もうひとつ[1]を入力。10進値は[511]。 	↓ もうひとつ[1]を入力。10進値は[-1]のまま(桁あふれ)。 

([0]と[1]が交互にくるような場合は異なる数値で正負が交互に逆転します。)

これを避ける方法を用意してもよいのですが、ここでは「補数表現で桁数を決めておく意味」が実際の入力でも理解されるよう、敢えて「正負が逆転する可能性」を残したままにしています。

補数表現での進法入力をおこなわれる際には、十分ご注意ください。

● 計算時の「端数処理」について

進法入力においても、もちろん「計算」は可能ですが、考えてみれば「四捨五入」というのはたとえば [0] と [1] だけの「2進数」においては、なかなか難しい問題となります。

2進数で計算した後に、暗黙裡に10進数に変換した上で四捨五入して、再び基数変換で2進数に戻す、というのも考えられなくはありませんが、基数変換に伴う循環小数の出現などを考慮すると、桁数の取り方などで値が変わってしまう可能性もあるため、それを電卓での計算結果とするのは、適当でないように思われます。

このことは「8進数」や「16進数」（もしくは他の基数）においても同様です。それは「四捨五入」という端数処理それ自体が [10] に満たない値を端数として「[4]まで」「[5]から」で「半分に分けよう」とする行為だからです。この「半分」の元になる「全体」= [10] というのは「十進法の基数」に他なりません。基数が違えば「半分」の捉え方も異なりますので、「四捨五入」の意味も、実は違うものになってきます。

ただし小数部の桁数を電卓画面で指定できるようにしておくことは、利便性の上で重要と考えられます。これらのことから当アプリでは、すべての「進法入力」における端数処理は指定桁数までの「切り捨て」のみとしております。

(このため進法入力時には上段側の「端数処理スライダ」は使えなくなります。下段側の「有効桁数スライダ」は使えます。)

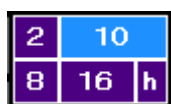
それ以外の「切り上げ」「四捨五入」「五捨五入」の端数処理が必要な場合、もしくは [flat] = 「端数処理なし」（設定画面で選択した「最大桁数」までの切り捨て）としたい場合は十進法の「通常入力」（進法入力のチェックを外した通常の入力方法）にてご利用ください。

● 次回起動時の選択状況

2・8・16進数の「進法入力」は、使いたいと思ったときだけ使えば良い、というものであって普段の使い始めはいつも安心できる10進数からでないと、混乱して不安に感じてしまうことでしょう。

したがいまして当アプリでは、次回の利用時に混乱してしまうのを避けるため電卓の起動時には必ず、十進法の「通常入力」から起動するようにしています。

これはすなわち、電卓画面左上の「進法入力」用の操作域でいいますと「基数選択」およびチェックボックスの「① 実施の可否」については今回の選択状況を 次回へ引き継がない ということを意味します。



基数選択



① ② ③

- | | | | |
|---|---------|------------|--------|
| ① | ・ 実施の可否 | 実際に進法入力を利用 | する／しない |
| ② | ・ 編集表示 | 設定画面の様式を適用 | する／しない |
| ③ | ・ 負数表現 | 正負の補数表現を適用 | する／しない |

なお、チェックボックスの ② ③ については、今回の状況を次回へも引き継ぎます。

また、設定画面の「進法入力」チェックボックスの状態も保存されますので、電卓画面においては「S」ボタンで切り替えた操作域が「進法入力」用なら、その状況も引き継がれます。

ただ「基数選択」と「① 実施の可否」についての状況は保存されないため、起動時にはともに初期値の「基数選択」=[10]、「① 実施の可否」=進法入力「しない」のままなので必ず十進法の「通常入力」から始められる、ということです。

さらに当アプリでは、チェックボックスの ② ③ の状況は引き継がれるため基数選択を[10]から[2][8][16]へと切り替えるという最小限の動作だけで「① 実施の可否」=進法入力「する」に切り替わります。

このため、すぐに前回までの「進法入力」の利用状況へと復帰させることが可能です。

「直算電卓」での「進法入力」

通常の電卓画面においては、2進数などの「進法入力」による「計算」をおこなっても単純に計算結果が出るだけでは、いまひとつ状況がわかりづらいため当アプリを作るにあたって、ひとつの目標にしたのがこの機能です。

(前述のように、使いたいと思ったときだけ使えば良い、という位置付けには変わりありませんし負の数の表現として「補数」を使う場合はおもに「加算」のみでそれ以外は無意味でもありますが)

上下2段の入力域で（基本的には）桁位置が揃うため、わかりやすくなっていると思われます。進法入力ですので、この場合も選択する基数に応じて、入力に使える数字（キー）が変わります。

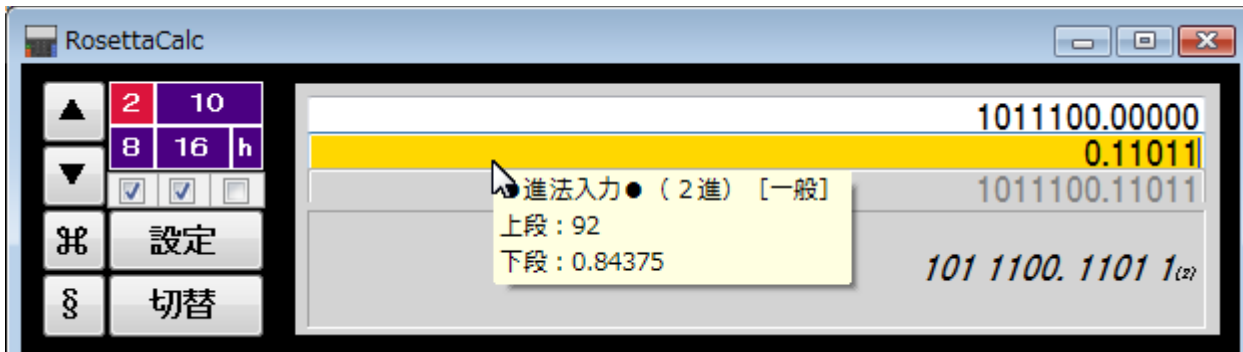
負数表現 [一般] で、正の数を加算 ([92] + [0.84375])。例示用に文字はやや大きくしています。(左上の画像は「進法入力」ではなく十進法の【通常入力】です。)



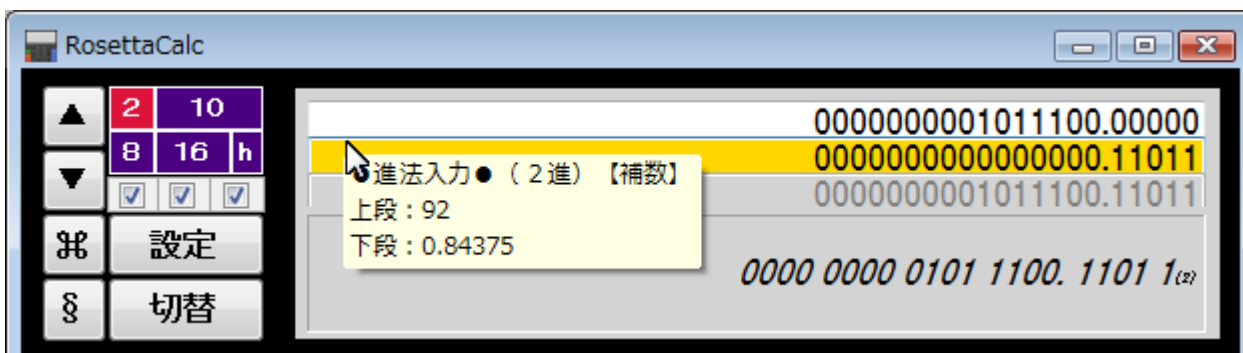
入力欄にマウスのポインタ（矢印、カレット）があるときには、ツールチップヒントが表示されますので入力欄の数値が、上下段ともに **10進数で表示** されて、つねに確認できます。

若干のわかりにくさを伴う「進法入力」での入力の進行中にも状況や数値を随時確認しながら作業を進めることができます。

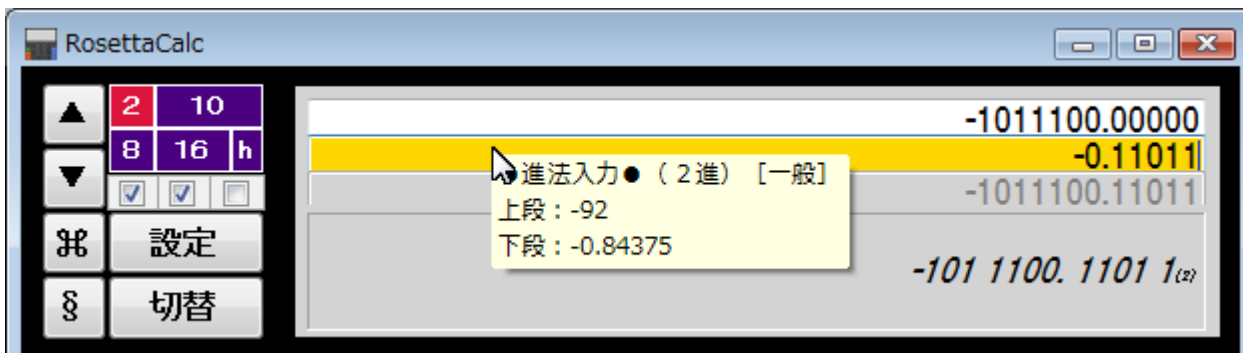
（例示：2進数の加算・桁区切り4桁・整数部16桁）



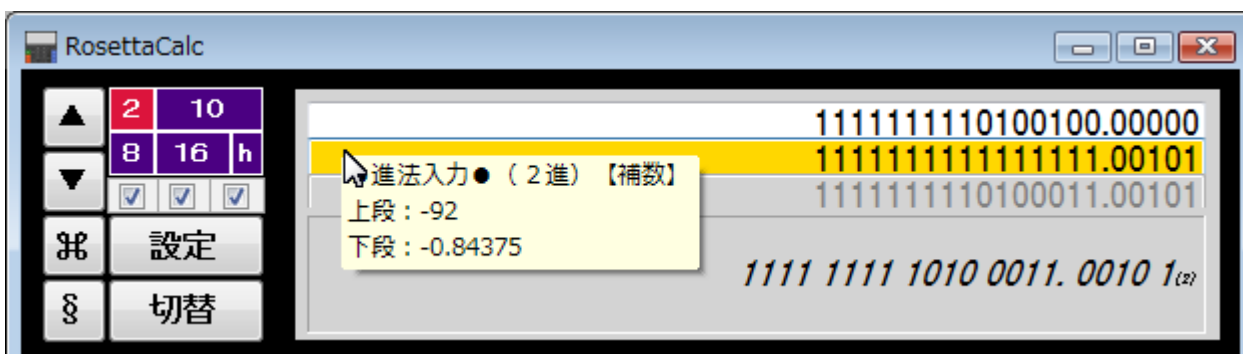
（前ページ画像の入力時の状況。正の2進数 [一般]）



（正の2進数【補数】）



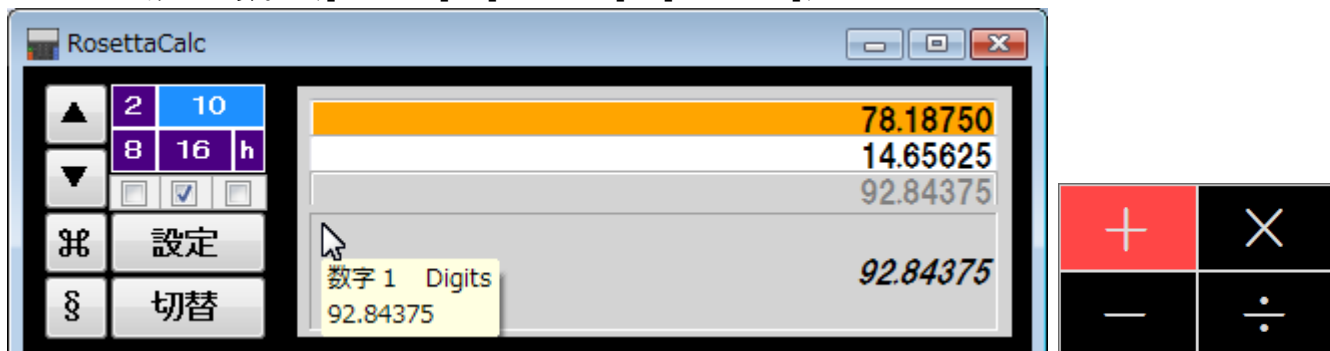
（負の2進数 [一般]）



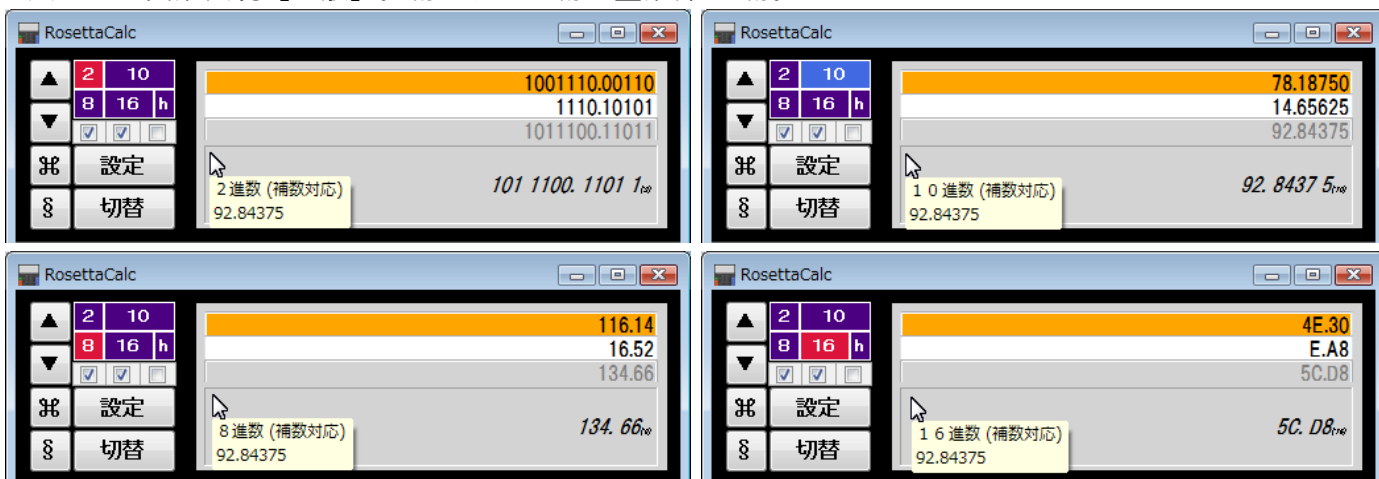
（負の2進数【補数】）

画像例 (すべて加算の場合で、他の基数についても例示。計算結果の数値は同じですが、元の数値は変えています。)

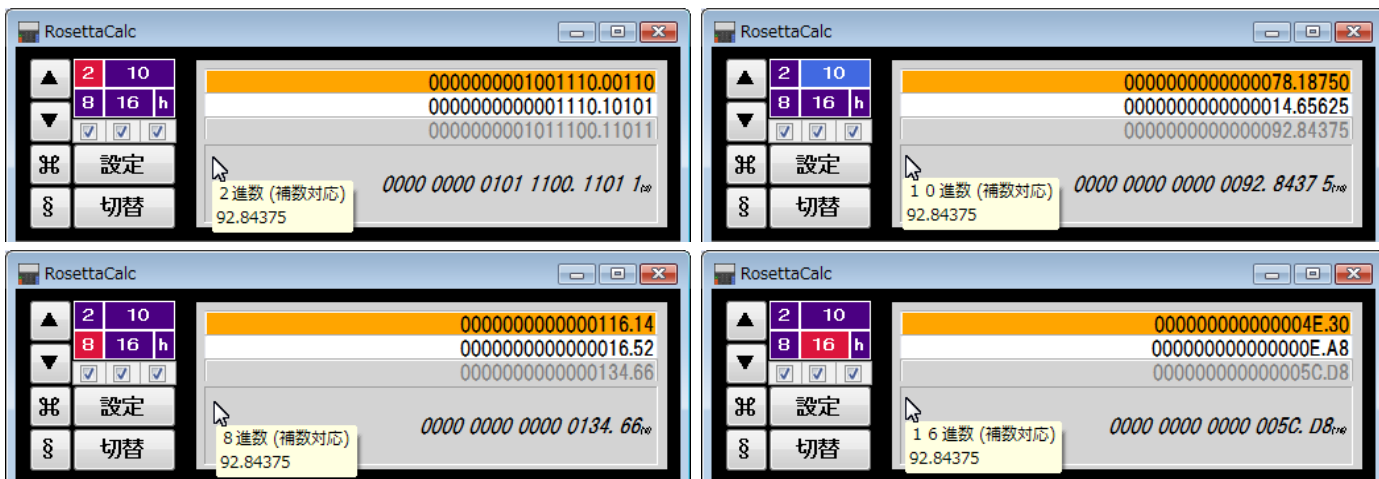
通常入力・正の数を加算。 ([78.1875] + [14.65625] = [92.84375])



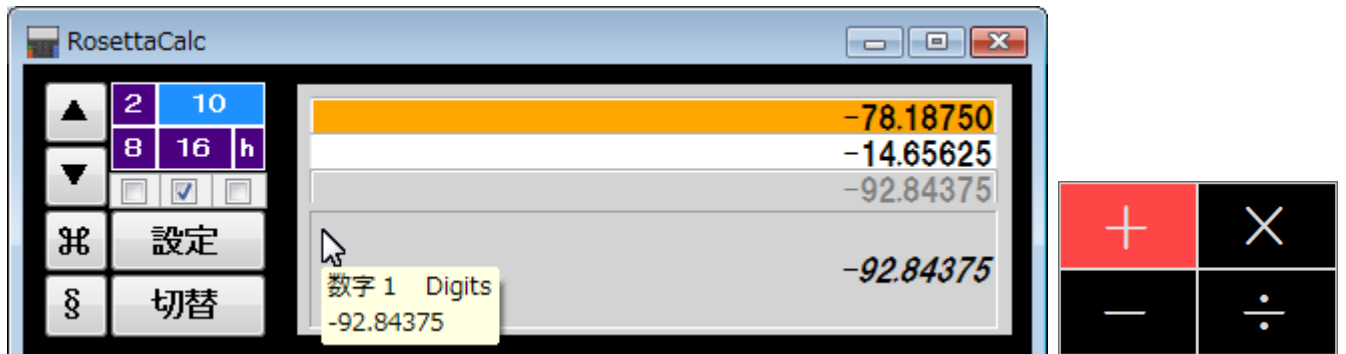
進法入力・負数表現 [一般]。桁区切り 4 桁・整数部 16 桁。



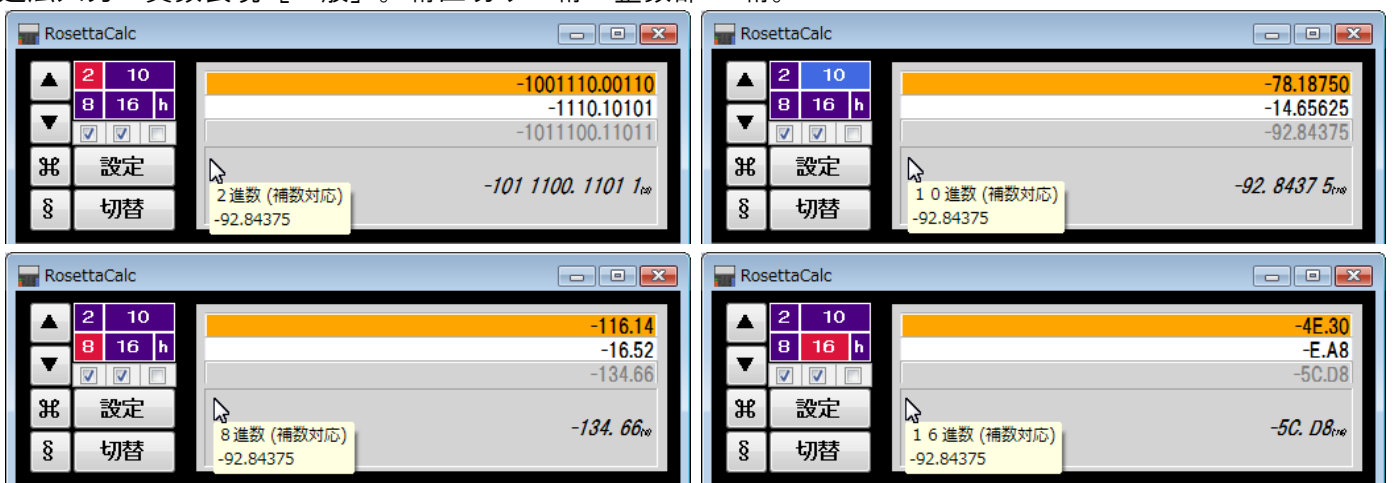
進法入力・負数表現 [補数]。桁区切り 4 桁・整数部 16 桁。



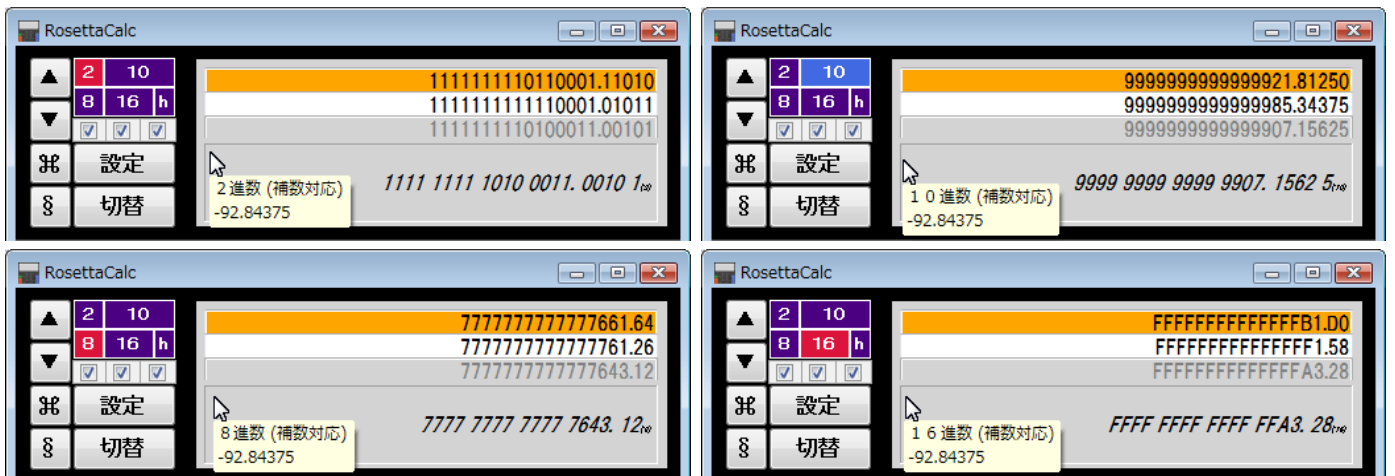
通常入力・負の数を加算。 $([-78.1875] + [-14.65625] = [-92.84375])$



進法入力・負数表現 [一般]。桁区切り 4 桁・整数部 16 桁。



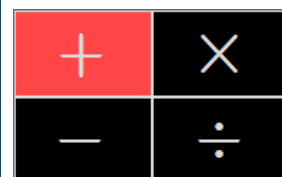
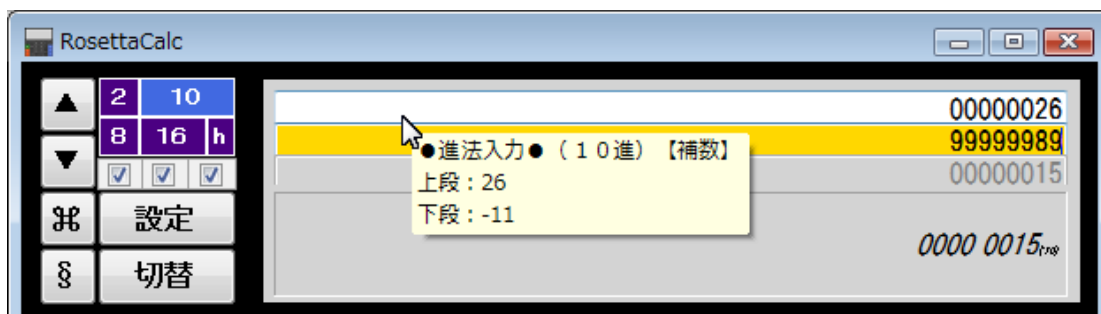
進法入力・負数表現 [補数]。桁区切り 4 桁・整数部 16 桁。



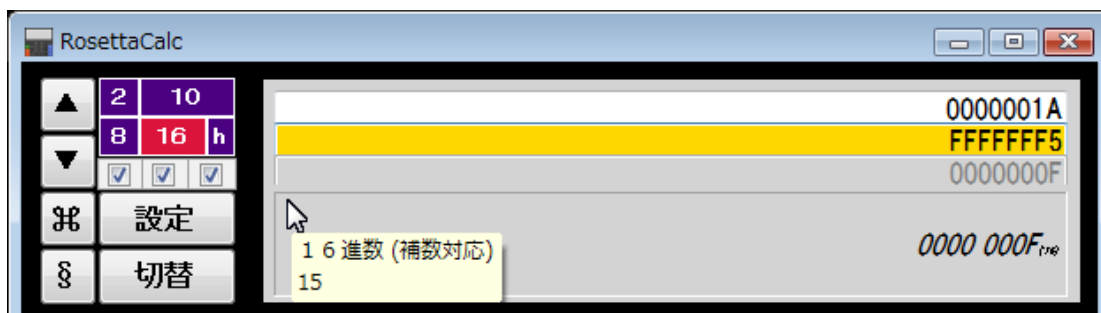
このように「直算電卓」において「進法入力」を使えば、基数法「n進数」での計算や負の数の表現に「補数」を使って「加算によって減算結果を得る」などの計算が手軽にわかりやすい形で、実際にやってみて体験できるようになります。

以下は前述の「補数について」の説明で例に挙げた「加算によって減算結果を得る」の計算です。

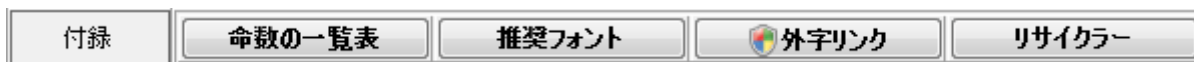
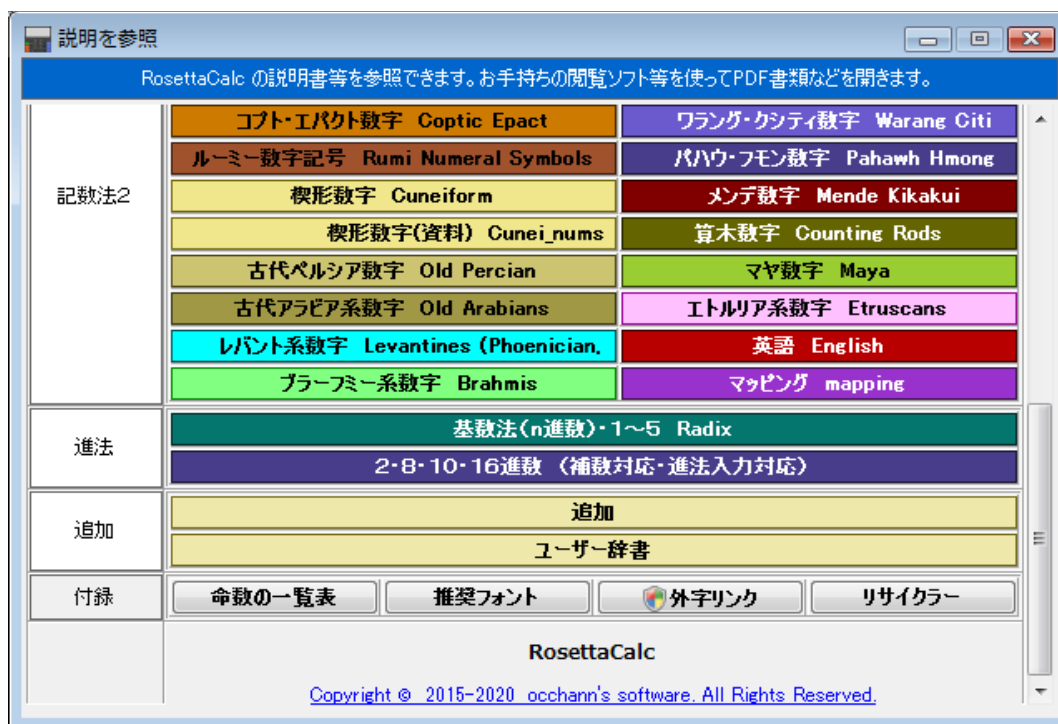
$$[0026] + [9989] = [10015] \quad ([26] + [-11] = [26] - [11] = [15])$$



	[26] n進法	減基数の補数	基数の補数	[11] n進法	減基数の補数	基数の補数
【2進数】	0001 1010	1110 0101	1110 0110	0000 1011	1111 0100	1111 0101
【8進数】	0000 0032	7777 7745	7777 7746	0000 0013	7777 7764	7777 7765
【10進数】	0000 0026	9999 9973	9999 9974	0000 0011	9999 9988	9999 9989
【16進数】	0000 001A	FFFF FFE5	FFFF FFE6	0000 000B	FFFF FFF4	FFFF FFF5



説明



電卓画面の [説明] ボタンから開くことのできる「説明を参照」画面で、後半部に「付録」としている [命数の一覧表] [推奨フォント] [外字リンク] [リサイクラー] の4項目は各ボタンをクリックすると、それぞれ付属のアプリ等が起動するようになっています。

「説明を参照」画面は「設定画面」よりも軽量なため、便利に手早く使える場合があります。普段は使わないことのほうが多いと思われませんが、必要に応じてご利用ください。

命数の一覧表

※ [命数の一覧表] は、当アプリ専用に付属しているものです。(設定画面からも起動できます。)
 世界各地で「位取り」などに使われる数値の呼び名「命数」を、参考のために言語や文化等でまとめたものです。
 (当方サイトにも同種のページを用意していますが、こちらはアプリとして動作するもので、小数部なども参照できます。)

RosettaCalc 命数の一覧表														
命数(大数)														
10のべき乗	漢数字	中数の万進法(4桁)	万進法(4桁)	万進法(4桁)の別名	ハングルの万進法(4桁)	ベトナムの千進法(3桁)	下数	上数	中数の万進法	高進	中数の万進法	仏典の語句	10のべき乗	アラビア数字千進法(3桁)
10 ⁰	一 (いち)	〇 一 二 三 四 五 六 七 八 九 十	一	一	一	一	一	一	一	一	一	一	10 ⁰	
10 ¹	十 (じゅう)		十	十	十	十	十	十	十	十	十	十	10 ¹	
10 ²	百 (ひゃく)		百	百	百	百	百	百	百	百	百	百	10 ²	
10 ³	千 (せん)		千	千	千	千	千	千	千	千	千	千	10 ³	
10 ⁴	万 (まん)		万	万	万	万	万	万	万	万	万	万	10 ⁴	
10 ⁵	十萬 (じゅうまん)		十萬	十萬	十萬	十萬	十萬	十萬	十萬	十萬	十萬	十萬	10 ⁵	
10 ⁶	百萬 (ひゃくまん)		百萬	百萬	百萬	百萬	百萬	百萬	百萬	百萬	百萬	百萬	10 ⁶	
10 ⁷	千萬 (せんまん)		千萬	千萬	千萬	千萬	千萬	千萬	千萬	千萬	千萬	千萬	10 ⁷	
10 ⁸	億 (億)		億	億	億	億	億	億	億	億	億	億	10 ⁸	
10 ⁹	十億 (じゅう億)		十億	十億	十億	十億	十億	十億	十億	十億	十億	十億	10 ⁹	
10 ¹⁰	百億 (ひゃく億)		百億	百億	百億	百億	百億	百億	百億	百億	百億	百億	10 ¹⁰	
10 ¹¹	千億 (せん億)		千億	千億	千億	千億	千億	千億	千億	千億	千億	千億	10 ¹¹	
10 ¹²	兆 (兆)		兆	兆	兆	兆	兆	兆	兆	兆	兆	兆	10 ¹²	
10 ¹³	十兆 (じゅう兆)		十兆	十兆	十兆	十兆	十兆	十兆	十兆	十兆	十兆	十兆	10 ¹³	
10 ¹⁴	百兆 (ひゃく兆)		百兆	百兆	百兆	百兆	百兆	百兆	百兆	百兆	百兆	百兆	10 ¹⁴	
10 ¹⁵	千兆 (せん兆)		千兆	千兆	千兆	千兆	千兆	千兆	千兆	千兆	千兆	千兆	10 ¹⁵	
10 ¹⁶	京 (京)		京	京	京	京	京	京	京	京	京	京	10 ¹⁶	
10 ¹⁷	十京 (じゅう京)		十京	十京	十京	十京	十京	十京	十京	十京	十京	十京	10 ¹⁷	
10 ¹⁸	百京 (ひゃく京)		百京	百京	百京	百京	百京	百京	百京	百京	百京	百京	10 ¹⁸	
10 ¹⁹	千京 (せん京)		千京	千京	千京	千京	千京	千京	千京	千京	千京	千京	10 ¹⁹	
10 ²⁰	垓 (垓)		垓	垓	垓	垓	垓	垓	垓	垓	垓	垓	10 ²⁰	
10 ²¹	十垓 (じゅう垓)		十垓	十垓	十垓	十垓	十垓	十垓	十垓	十垓	十垓	十垓	10 ²¹	
10 ²²	百垓 (ひゃく垓)		百垓	百垓	百垓	百垓	百垓	百垓	百垓	百垓	百垓	百垓	10 ²²	
10 ²³	千垓 (せん垓)		千垓	千垓	千垓	千垓	千垓	千垓	千垓	千垓	千垓	千垓	10 ²³	
10 ²⁴	穠 (穠)		穠	穠	穠	穠	穠	穠	穠	穠	穠	穠	10 ²⁴	
10 ²⁵	十穠 (じゅう穠)		十穠	十穠	十穠	十穠	十穠	十穠	十穠	十穠	十穠	十穠	10 ²⁵	
10 ²⁶	百穠 (ひゃく穠)		百穠	百穠	百穠	百穠	百穠	百穠	百穠	百穠	百穠	百穠	10 ²⁶	
10 ²⁷	千穠 (せん穠)		千穠	千穠	千穠	千穠	千穠	千穠	千穠	千穠	千穠	千穠	10 ²⁷	
10 ²⁸	穰 (穰)		穰	穰	穰	穰	穰	穰	穰	穰	穰	穰	10 ²⁸	
10 ²⁹	十穰 (じゅう穰)		十穰	十穰	十穰	十穰	十穰	十穰	十穰	十穰	十穰	十穰	10 ²⁹	
10 ³⁰	百穰 (ひゃく穰)		百穰	百穰	百穰	百穰	百穰	百穰	百穰	百穰	百穰	百穰	10 ³⁰	
10 ³¹	千穰 (せん穰)		千穰	千穰	千穰	千穰	千穰	千穰	千穰	千穰	千穰	千穰	10 ³¹	

(このアプリは表示項目が多いため、表示を最大化したほうが使いやすくなります。)

推奨フォント

※ [推奨フォント] ボタンから起動されるのは、当方作成のフリーソフト「推奨フォントのダウンローダ」(RcmFntDLDer)です。電卓での特殊な数値表現に欠かせない、当方推奨の各種フォントをまとめてダウンロード・インストールできます。(このフリーソフトは当アプリには付属していませんので、ご利用には別途、事前のインストールが必要です。)

font 推奨フォントのダウンローダ

広範な Unicode 用の文字の多くを表示するために有用な、できるだけ最新のフォントをダウンロードします。

すべて
 推奨フォント以外もすべて表示
 推奨フォント(1次2次)を表示
 推奨フォント(1次のみ)を表示

当方作成アプリ推奨フォント
 UniCharFinder RosettaCalc

状況確認1
 作業用フォルダを指定

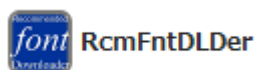
状況確認2
 再確認
 未登録にチェック

対象	フォント名	作成/提供者名	圧縮	Site	説明	page	pdf	個別DL
<input checked="" type="checkbox"/>	Abyssinica SIL	SIL International	zip	Go	Go	Go	Go	<DL>
<input type="checkbox"/>	Aegean	Abyssinica SIL						<DL>
<input type="checkbox"/>	Aegyptus							<DL>
<input type="checkbox"/>	AegyptusS							<DL>
<input type="checkbox"/>	AhomUnico							<DL>
<input type="checkbox"/>	Akkadian							<DL>
<input type="checkbox"/>	Albanian	H.Krasniqi	zip	Go	Go	Go	Go	<DL>

処理を選択してください
 下で選択した、一連または単独の処理をおこないます。ダウンロードをおこなうなどの場合は、上の表で対象を選択してください。 [Noto Fonts](#)
 上の表で選択できない行は、システムに付属しプリインストールされる「標準フォント」、または「シェアウェア」などの、自動処理に馴染まないものです。

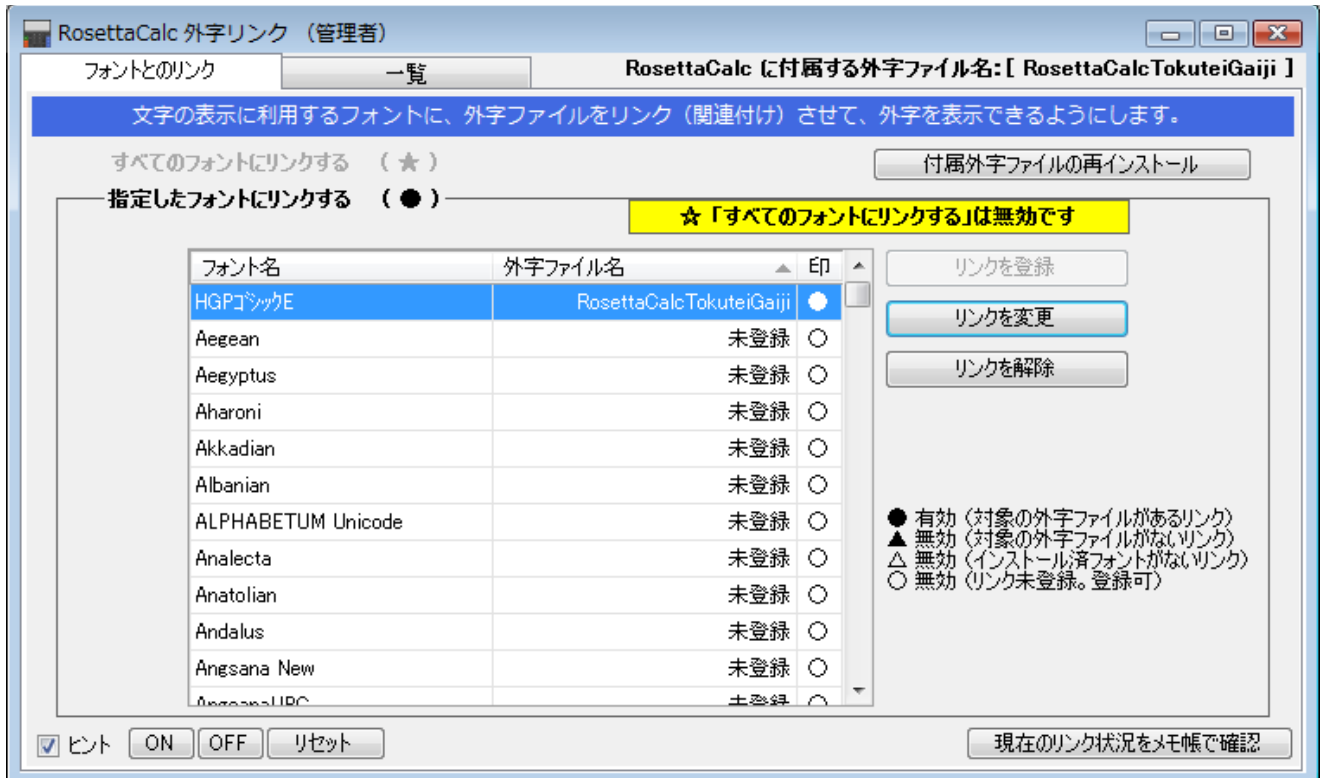
<input type="checkbox"/> 選択したフォントを、作業用フォルダへ 自動ダウンロードする	<input type="checkbox"/> 選択したフォントに別添の pdf 説明書を、自動ダウンロードする
<input type="checkbox"/> 作業用フォルダ内の 圧縮ファイルを、すべて解凍する	<input type="checkbox"/> すべて予備で代用 (occhann.jp)
<input type="checkbox"/> 作業用フォルダ内の 圧縮ファイルを、解凍できた都度、削除する	<input checked="" type="checkbox"/> 終了後、作業用フォルダとログを開く
<input type="checkbox"/> 作業用フォルダ内の 選択したフォントを、自動インストールする	
<input type="checkbox"/> 作業用フォルダ内の すべてのフォントを、自動インストールする	

処理を選択してください



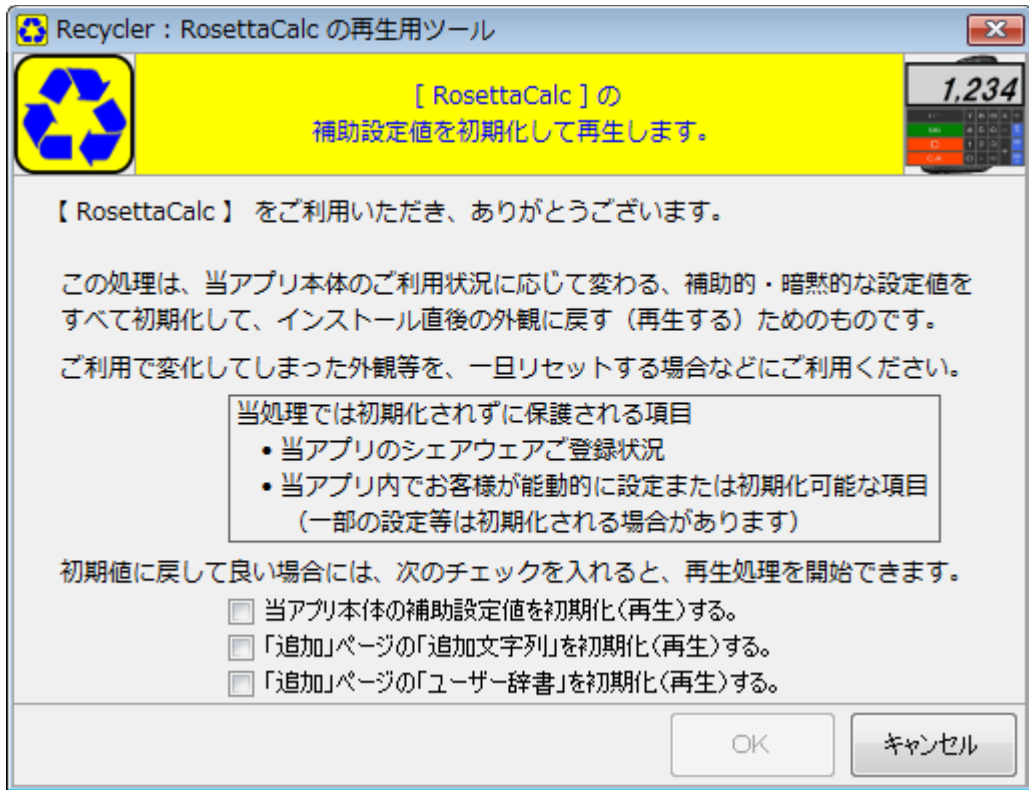
外字リンク

※ [外字リンク] アプリは、当アプリ専用に付属しており、「外字」をフォント経由で表示できるように設定するためのものです。外字ファイルとフォントとのリンク(関連付け)の設定は、システムのレジストリに登録されるようになっておりますのでこれを扱うため、起動時には「管理者権限の承認」が求められます。(このアプリは設定画面からも起動できます。)

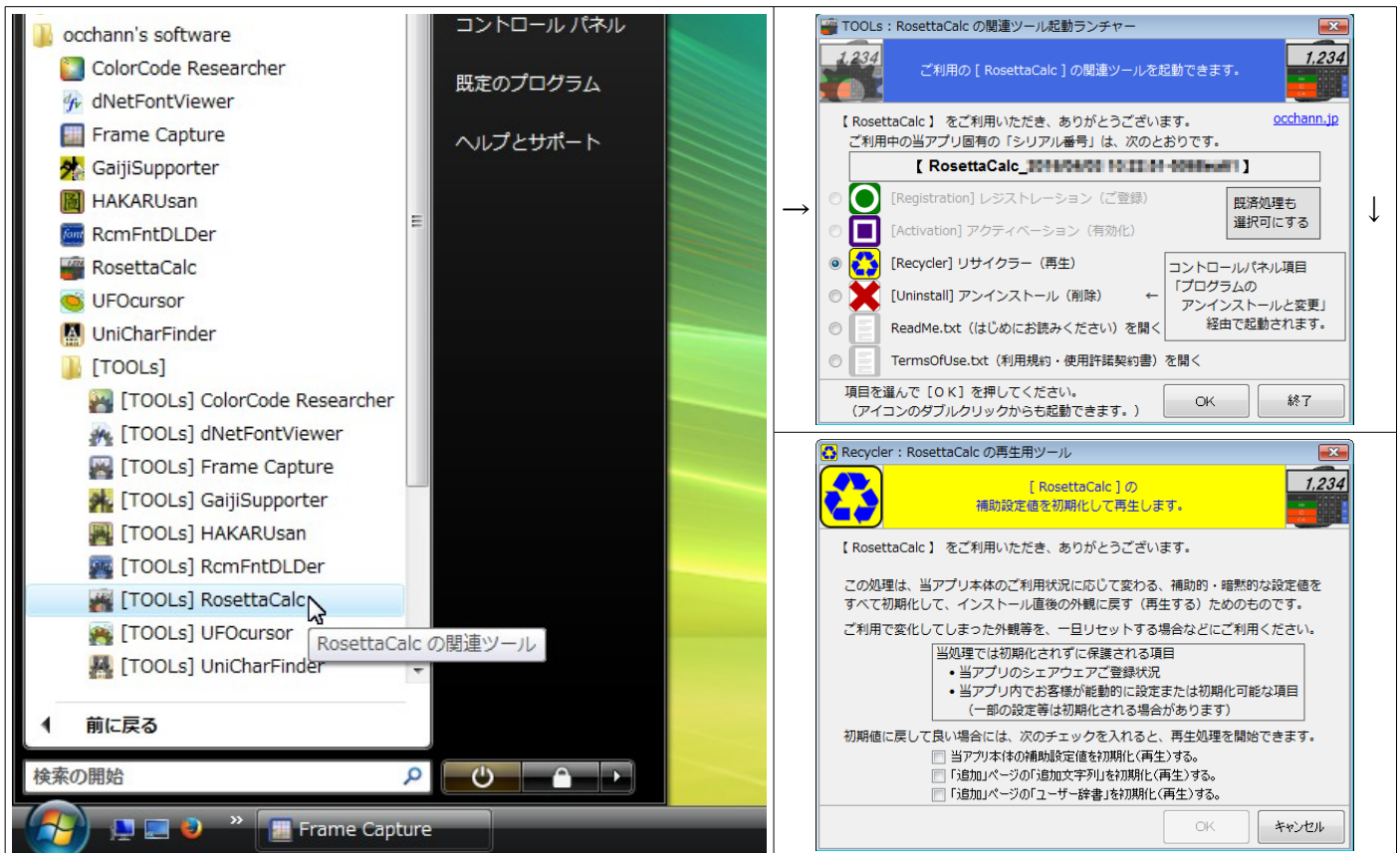


リサイクラー

※ [リサイクラー] は、当アプリ専用に付属しており、一般にはスタートボタン経由の[T00Ls]メニューから起動されるものです。当アプリの各種設定をインストール直後の状態に「リセット」したい場合に使えます。(リセット機能は設定画面にもチェックボックスの指定で使える類似機能を用意しています。)



なお、当方作成の各種アプリで標準としている「関連ツール」は、下図のように、「スタート」ボタンを押すと表示される「すべてのプログラム」から見える「occhann's software」フォルダ内の[T00Ls]フォルダから開くようになっています。





RosettaCalc
ver. 1.06
電卓画面 Calculator

[Copyright © 2015-2020 occhann's software. All Rights Reserved.](#)