Calcpad

Version 6.0 quick reference guide

ПРОЕКТСОФТ®
34-36 Peyo Yavorov blvd, Sofia 1164, Bulgaria
+359 2 423 4455
proektsoft.bg@gmail.com

How it works?

1. **Enter** text and formulas into the "**Code**" box on the left.
2. Press **F5** or click ▶ to **calculate**. The results will appear in the "**Output**" box on the right as a professionally formatted Html **report**.
3. Click 🖶 to **print** or 📄 to **copy** the output.
   You can also **export** it to **Html** 💾, **PDF** 📕 or **MS Word** 📝 document.

The language

The Calcpad language includes the following elements:

- Real numbers: digits "**0**" - "**9**" and decimal point "**.**";
- Complex numbers: **re** ± **im***i* (e.g. **3** - **2***i*);
- Variables:
  - Latin letters: a - z, A - Z;
  - Greek letters: α - ω, A - Ω;
  - digits: **0** - **9**;
  - comma: " **,** ";
  - prime symbols: ′ , ″ , ‴ , ⁗ ;
  - superscripts: ⁰ , ¹ , ² , ³ , ⁴ , ⁵ , ⁶ , ⁷ , ⁸ , ⁹ , ⁿ , ⁺ , ⁻ ;
  - special symbols: ¯ , ø , Ø , ° , ∡ ;
  - " **_** " for subscript;

  A variable name must start with a letter. Names are case sensitive.
- Constants: π, e, φ, γ, g, G, M_E, M_S, c, h, μ_0, ε_0, k_e, e, m_e, m_p, m_n, N_A, σ, k_B, R, F, γ_c, γ_s, γ_a, γ_g, γ_w
- Operators:
  "**!**" - factorial;
  "**^**" - exponent;
  "**/**" - division;
  "**÷**" - force division bar;
  "****" - division;
  "**⊘**" - modulo (reminder);
  "***** " - multiplication;
  "**-**" - minus;
  "**+**" - plus;
  "**≡**" - equal to;
  "**≠**" - not equal to;
  "**<**" - less than;
  "**>**" - greater than;
  "**≤**" - less or equal;

"≥" - greater or equal;

"∧" - logical “and”;

"∨" - logical “or”;

"⊕" - logical “xor”;

"=" - assignment;

- Custom functions of type $f(x; y; z; ...)$;
- Built-in functions:
  - Trigonometric:

    $\mathbf{sin}(x)$ - sine;

    $\mathbf{cos}(x)$ - cosine;

    $\mathbf{tan}(x)$ - tangent;

    $\mathbf{csc}(x)$ - cosecant;

    $\mathbf{sec}(x)$ - secant;

    $\mathbf{cot}(x)$ - cotangent;
  - Hyperbolic:

    $\mathbf{sinh}(x)$ - hyperbolic sine;

    $\mathbf{cosh}(x)$ - hyperbolic cosine;

    $\mathbf{tanh}(x)$ - hyperbolic tangent;

    $\mathbf{csch}(x)$ - hyperbolic cosecant;

    $\mathbf{sech}(x)$ - hyperbolic secant;

    $\mathbf{coth}(x)$ - hyperbolic cotangent;
  - Inverse trigonometric:

    $\mathbf{asin}(x)$ - inverse sine;

    $\mathbf{acos}(x)$ - inverse cosine;

    $\mathbf{atan}(x)$ - inverse tangent;

    $\mathbf{atan2}(x; y)$ - the angle whose tangent is the quotient of y and x;

    $\mathbf{acsc}(x)$ - inverse cosecant;

    $\mathbf{asec}(x)$ - inverse secant;

    $\mathbf{acot}(x)$ - inverse cotangent;
  - Inverse hyperbolic:

    $\mathbf{asinh}(x)$ inverse hyperbolic sine;

    $\mathbf{acosh}(x)$ - inverse hyperbolic cosine;

    $\mathbf{atanh}(x)$ - inverse hyperbolic tangent;

    $\mathbf{acsch}(x)$ - inverse hyperbolic cosecant;

    $\mathbf{asech}(x)$ - inverse hyperbolic secant;

    $\mathbf{acoth}(x)$ - inverse hyperbolic cotangent;
  - Logarithmic, exponential and roots:

    $\mathbf{log}(x)$ - decimal logarithm;

    $\mathbf{ln}(x)$ - natural logarithm;

log_2(*x*) - binary logarithm;

exp(*x*) – exponential function;

sqr(*x*) or **sqrt**(*x*) - square root;

cbrt(*x*) - cubic root;

root(*x*; *n*) - n-th root;

◦ Rounding:

round(*x*) - round to the nearest integer;

floor(*x*) - round to the smaller integer (towards -∞);

ceiling(*x*) - round to the greater integer (towards +∞);

trunc(*x*) - round to the smaller integer (towards zero);

◦ Integer:

mod(*x*; *y*) - the reminder of an integer division;

gcd(*x*; *y*) - the greatest common divisor of two integers;

lcm(*x*; *y*) - the least common multiple of two integers;

◦ Complex:

abs(*x*) - absolute value/magnitude;

re(*x*) - the real part of a complex number;

im(*x*) - the imaginary part of a complex number;

phase(*x*) - the phase of a complex number;

◦ Aggregate and interpolation:

min(*x*; *y*; *z*...) - minimum of multiple values;

max(*x*; *y*; *z*...) - maximum of multiple values;

sum(*x*; *y*; *z*...) - sum of multiple values = $x + y + z$...;

sumsq(*x*; *y*; *z*...) - sum of squares = $x^2 + y^2 + z^2$...;

srss(*x*; *y*; *z*...) - square root of sum of squares = **sqrt**($x^2 + y^2 + z^2$...);

average(*x*; *y*; *z*...) - average of multiple values = $(x + y + z...)/n$;

product(*x*; *y*; *z*...) - product of multiple values = $x \cdot y \cdot z$...;

mean(*x*; *y*; *z*...) - geometric mean = **n-th root**($x \cdot y \cdot z$...);

take(*n*; *a*; *b*; *c*...) - returns the n-th element from the list;

line(*x*; *a*; *b*; *c*...) - linear interpolation;

spline(*x*; *a*; *b*; *c*...) - Hermite spline interpolation;

◦ Conditional and logical:

if(*cond*; *value-if-true*; *value-if-false*) - conditional evaluation;

switch(*cond1*; *value1*; *cond2*; *value2*; ...; *default*) - selective evaluation;

not(*x*) - logical "not";

and(*x*; *y*; *z*...) - logical "and";

or(*x*; *y*; *z*...) - logical "or";

xor(*x*; *y*; *z*...) - logical "xor";

◦ Other:

sign(x) - sign of a number;

random(x) - random number between 0 and x.

- Comments: **"Title"** or **'text'** in double or single quotes, respectively. **HTML**, **CSS**, **JS** and **SVG** are allowed.
- Graphing and plotting:

  $Plot { $\boldsymbol{f}(x)$ @ $x = a : b$ } - simple plot;

  $Plot { $\boldsymbol{x}(t) \mid \boldsymbol{y}(t)$ @ $t = a : b$ } - parametric;

  $Plot { $\boldsymbol{f}_1(x)$ & $\boldsymbol{f}_2(x)$ & ... @ $x = a : b$ } - multiple;

  $Plot { $\boldsymbol{x}_1(t) \mid \boldsymbol{y}_1(t)$ & $\boldsymbol{x}_2(t) \mid \boldsymbol{y}_2(t)$ & ... @ $x = a : b$ } - multiple parametric;

  $Map { $\boldsymbol{f}(x; y)$ @ $x = a : b$ & $y = c : d$ } - 2D color map of a 3D surface;

  PlotHeight - height of plot area in pixels;

  PlotWidth - width of plot area in pixels;

  PlotStep - grid size for map plotting.

- Iterative and numerical methods:

  $Root { $\boldsymbol{f}(x) = const$ @ $x = a : b$ } - root finding for $\boldsymbol{f}(x) = const$;

  $Root { $\boldsymbol{f}(x)$ @ $x = a : b$ } - root finding for $\boldsymbol{f}(x) = 0$;

  $Find { $\boldsymbol{f}(x)$ @ $x = a : b$ } similar to above, but x is not required to be a precise solution;

  $Sup { $\boldsymbol{f}(x)$ @ $x = a : b$ } - local maximum of a function;

  $Inf { $\boldsymbol{f}(x)$ @ $x = a : b$ } - local minimum of a function;

  $Area { $\boldsymbol{f}(x)$ @ $x = a : b$ } - adaptive Gauss-Lobatto numerical integration;

  $Integral { $\boldsymbol{f}(x)$ @ $x = a : b$ } – Tanh-Sinh numerical integration;

  $Slope { $\boldsymbol{f}(x)$ @ $x = a$ } - numerical differentiation;

  $Sum { $\boldsymbol{f}(x)$ @ $k = a : b$ } - iterative sum;

  $Product { $\boldsymbol{f}(k)$ @ $k = a : b$ } - iterative product;

  $Repeat { $\boldsymbol{f}(k)$ @ $k = a : b$ } - general inline iterative procedure;

  Precision - relative precision for numerical methods [10^{-2}; 10^{-16}] (default is 10^{-12})

- Program flow control:

  Simple:

  ```
  #if condition
      your code goes here
  #end if
  ```

  Alternative:

  ```
  #if condition
      your code goes here
  #else
      some other code
  #end if
  ```

  Complete:

  ```
  #if condition1
  ```

```
    your code goes here
#else if condition2
    your code goes here
#else
    some other code
#end if
```

You can add as many "#else if"s as needed, but only one "#else". You can omit any of them.

- Iteration blocks:

  Simple:

```
#repeat number of repetitions
    your code goes here
#loop
```

  With conditional break/continue:

```
#repeat number of repetitions
    your code goes here
    #if condition
        #break or #continue
    #end if
    some more code
#loop
```

- Modules and macros/string variables:

  Modules:

  #include *filename* - include external file (module);

  #local - start local section (not to be included);

  #global - start global section (to be included);

  Inline string variable:

```
#def variable_name$ = content
```

  Multiline string variable:

```
#def variable_name$
    content line 1
    content line 2
    ...
#end def
```

  Inline string macro:

```
#def macro_name$(param1$; param2$; ...) = content
```

  Multiline string macro:

```
#def macro_name$(param1$; param2$; ...)
    content line 1
    content line 2
    ...
#end def
```

- Output control:
  #hide - hide the report contents;
  #show - always show the contents (default);
  #pre - show the next contents only before calculations;
  #post - show the next contents only after calculations;
  #val - show only the final result, without the equation;
  #equ - show complete equations and results (default);
  #noc - show only equations without results (no calculations);
  #round *n* - rounds the output to *n* digits after the decimal point.
  Each of the above commands is effective after the current line until the end of the report or another command that overwrites it.
- Breakpoints for step-by-step execution:
  #pause - calculates down to the current line and waits for the user to resume manually;
  #input - renders an input form to the current line and waits for user input.
- Units for trigonometric functions: #deg - degrees, #rad - radians, #gra – grades;
- Separator for target units: |;
- Return angles with units: *ReturnAngleUnits* = 1;
- Dimensionless units: *%*, *‰*;
- Angle units *°*, *′*, *″*, *deg*, *rad*, *grad*, *rev*;
- Metric units (SI and compatible):
  Mass: *g*, *hg*, *kg*, *t*, *kt*, *Mt*, *Gt*, *dg*, *cg*, *mg*, *µg*, *Da*, *u*;
  Length: *m*, *km*, *dm*, *cm*, *mm*, *µm*, *nm*, *pm*, *AU*, *ly*;
  Time: *s*, *ms*, *µs*, *ns*, *ps*, *min*, *h*, *d*;
  Frequency: *Hz*, *kHz*, *MHz*, *GHz*, *THz*, *mHz*, *µHz*, *nHz*, *pHz*, *rpm*;
  Speed: *kmh*;
  Electric current: *A*, *kA*, *MA*, *GA*, *TA*, *mA*, *µA*, *nA*, *pA*;
  Temperature: *°C*, *Δ°C*, *K*;
  Amount of substance: *mol*;
  Luminous intensity: *cd*;
  Area: *a*, *daa*, *ha*;
  Volume: *L*, *daL*, *hL*, *dL*, *cL*, *mL*, *µL*, *nL*, *pL*;
  Force: *dyn N*, *daN*, *hN*, *kN*, *MN*, *GN*, *TN*, *gf*, *kgf*, *tf*;
  Moment: *Nm*, *kNm*;
  Pressure: *Pa*, *daPa*, *hPa*, *kPa*, *MPa*, *GPa*, *TPa*,
  dPa, *cPa*, *mPa*, *µPa*, *nPa*, *pPa*,
  bar, *mbar*, *µbar*, *atm*, *at*, *Torr*, *mmHg*;
  Viscosity: *P*, *cP*, *St*, *cSt*;
  Energy work: *J*, *kJ*, *MJ*, *GJ*, *TJ*, *mJ*, *µJ*, *nJ*, *pJ*,
  Wh, *kWh*, *MWh*, *GWh*, *TWh*, *cal*, *kcal*, *erg*,
  eV, *keV*, *MeV*, *GeV*, *TeV*, *PeV*, *EeV*;

Power: *W*, *kW*, *MW*, *GW*, *TW*, *mW*, *µW*, *nW*, *pW*, *hpM*, *ks*,
VA, *kVA*, *MVA*, *GVA*, *TVA*, *mVA*, *µVA*, *nVA*, *pVA*,
VAR, *kVAR*, *MVAR*, *GVAR*, *TVAR*, *mVAR*, *µVAR*, *nVAR*, *pVAR*;
Electric charge: *C*, *kC*, *MC*, *GC*, *TC*, *mC*, *µC*, *nC*, *pC*, *Ah*, *mAh*;
Potential: *V*, *kV*, *MV*, *GV*, *TV*, *mV*, *µV*, *nV*, *pV*;
Capacitance: *F*, *kF*, *MF*, *GF*, *TF*, *mF*, *µF*, *nF*, *pF*;
Resistance: *Ω*, *kΩ*, *MΩ*, *GΩ*, *TΩ*, *mΩ*, *µΩ*, *nΩ*, *pΩ*;
Conductance: *S*, *kS*, *MS*, *GS*, *TS*, *mS*, *µS*, *nS*, *pS*,
℧, *k℧*, *M℧*, *G℧*, *T℧*, *m℧*, *µ℧*, *n℧*, *p℧*;
Magnetic flux: *Wb* , *kWb*, *MWb*, *GWb*, *TWb*, *mWb*, *µWb*, *nWb*, *pWb*;
Magnetic flux density: *T*, *kT*, *MT*, *GT*, *TT*, *mT*, *µT*, *nT*, *pT*;
Inductance: *H*, *kH*, *MH*, *GH*, *TH*, *mH*, *µH*, *nH*, *pH*;
Luminous flux: *lm*;
Illuminance: *lx*;
Radioactivity: *Bq*, *kBq*, *MBq*, *GBq*, *TBq*, *mBq*, *µBq*, *nBq*, *pBq*, *Ci*, *Rd*;
Absorbed dose: *Gy*, *kGy*, *MGy*, *GGy*, *TGy*, *mGy*, *µGy*, *nGy*, *pGy*;
Equivalent dose: *Sv*, *kSv*, *MSv*, *GSv*, *TSv*, *mSv*, *µSv*, *nSv*, *pSv*;
Catalytic activity: *kat*;

- Non-metric units (Imperial/US):
  Mass: *gr*, *dr*, *oz*, *lb* (or *lbm*, *lb_m*), *kipm* (or *kip_m*), *st*, *qr*,
  cwt (or *cwt_uk*, *cwt_us*), *ton* (or *ton_uk*, *ton_us*), *slug*;
  Length: *th*, *in*, *ft*, *yd*, *ch*, *fur*, *mi*, *ftm* (or *ftm_uk*, *ftm_us*),
  cable (or *cable _uk*, *cable _us*), *nmi*, *li*, *rod*, *pole*, *perch*, *lea*;
  Speed: *mph*, *knot*;
  Temperature: *°F*, *Δ°F*, *°R*;s
  Area: *rood*, *ac*;
  Volume, fluid: *fl_oz*, *gi*, *pt*, *qt*, *gal*, *bbl*, or:
  fl_oz_uk, *gi_uk*, *pt_uk*, *qt_uk*, *gal_uk*, *bbl_uk*,
  fl_oz_us, *gi_us*, *pt_us*, *qt_us*, *gal_us*, *bbl_us*;
  Volume, dry: (US) *pt_dry*, (US) *qt_dry*, (US) *gal_dry* (US) *bbl_dry*,
  pk (or *pk_uk*, *pk_us*), *bu* (or *bu_uk*, *bu_us*);
  Force: *ozf* (or *oz_f*), *lbf* (or *lb_f*), *kip* (or *kipf*, *kip_f*), *tonf* (or *ton_f*), *pdl*;
  Pressure: *osi*, *osf psi*, *psf*, *ksi*, *ksf*, *tsi*, *tsf*, *inHg*;
  Energy/work: *BTU*, *therm* (or *therm_uk*, *therm_us*), *quad*;
  Power: *hp*, *hpE*, *hpS*;
- Custom units - *.Name* = expression.
  Names can contain currency symbols: €, £, ₤, ¥, ¢, ₽, ₹, ₩, ₪.