Qualculate Web

Manvith, Shreyas, Ashwin G Rao, Karthik Wagle

June 8, 2024

1 Introduction

Qalculate! is a multi-purpose cross-platform desktop calculator, But It's functionalities are mostly limited to offline. It is one of the most powerful calculators which calculate almost any problem. To use this on a web interface we are using php. PHP has good support for server side scripting using which applicatons are ran on web.

The Basic versions of Qalculate have command line interface, gtk and qt versions for GUI. For web we are usig commandline and gtk versions for displaying graphs and getting results.

2 Supported Unit Lists

a) Angle

- Plane Angle
- Solid Angle

b) Area

- Currency Electricity
- Capacitance
- Electric Charge
- Electric Conductance
- Electric Current
- Electric Dipole Moment
- Electric Potential
- Electric Resistance
- Electrical Elastance
- Inductance

c) Energy

- Action
- Entropy
- Power

d) Force

- Dynamic Viscosity
- Kinematic Viscosity
- Pressure

e) Information

f) Length Light

- Illuminance
- Luminance
- Luminous Flux
- Luminous Intensity

g) Magnetism

- Magnetic Field Strength
- Magnetic Flux
- Magnetic Flux Density

h) Mass Radioactivity

- Absorbed Dose

- Dose Equivalent Exposure

i) Ratio Speed

- Acceleration

j) Substance

- Catalytic Activity

k) Temperature Time

- Frequency

l) Typography Volume

- Cooking
- Fuel Economy
- Imperial Capacity
- U.S. Capacity
- Volumetric Flow Rate

3 Capabilities

a) Algebra

b) Bitwiser Operatioans

c) Calculus

d) Combinatorics

e) Complex Numbers

f) Currency Conversion

g) Data sets

h) Date and Time

i) Economics

j) Geometry

- Circle
- Cone
- Cube
- Cylinder
- Parallelogram
- Prism
- Pyramid
- Rectangle
- Sphere
- Square
- Trapezoid
- Triangle

k) Logical Operations

l) Matrices and Vectors

m) Number Conversions

- Arithmetic
- Integers
- Number Bases
- Polynomials
- Prime Numbers
- Rounding

n) Plotting a function

o) Plotting Graphs

p) Relational Operations

q) Statistics

- Correlation
- Descriptive Statistics
- Distribution
- Means
- Moments
- Random Numbers
- Regression
- Statistical Tests

r) Trignometry

s) Unit Conversions

Example

a) Basic functions and operators

- sqrt 4 = sqrt(4) = 4^(0.5) = 4^(1/2) = 2
- sqrt(25; 16; 9; 4) = [5 4 3 2]
- sqrt(32) = 4 × √(2) (in exact mode)
- cbrt(-27) = root(-27; 3) = -3 (real root)
- (-27)^(1/3) ≈ 1.5 + 2.5980762i (principal root)
- ln 25 = log(25; e) ≈ 3.2188758
- log2(4)/log10(100) = log(4; 2)/log(100; 10) = 1
- 5! = 1 × 2 × 3 × 4 × 5 = 120
- 5\2 = 5//2 = trunc(5 / 2) = 2 (integer division)
- 5 mod 3 = mod(5; 3) = 2
- 52 to factors = 2^2 × 13
- 25/4 × 3/5 to fraction = 3 + 3/4
- gcd(63; 27) = 9
- sin(pi/2) - cos(pi) = sin(90 deg) - cos(180 deg) = 2
- sum(x; 1; 5) = 1 + 2 + 3 + 4 + 5 = 15
- sum(\i^2+sin(\i); 1; 5; \i) = 1^2 + sin(1) + 2^2 + sin(2) + ... ≈ 55.176162
- product(x; 1; 5) = 1 × 2 × 3 × 4 × 5 = 120
- var1:=5 (stores value 5 in variable var1)
- var1 × 2 = 10
- 5^2 #this is a comment = 25
- sinh(0.5) where sinh()=cosh() = cosh(0.5) ≈ 1.1276260
- plot(x^2; -5; 5) (plots the function y=x^2 from -5 to 5)

b) Units

- 5 dm3 to L = 5 dm^3 to L = 5 L
- 20 miles / 2h to km/h = 16.09344 km/h
- 1.74 to ft = 1.74 m to ft ≈ 5 ft + 8.5039370 in
- 1.74 m to -ft ≈ 5.7086614 ft
- 100 lbf × 60 mph to hp ≈ 16 hp
- 50 Ω × 2 A = 100 V
- 50 Ω × 2 A to base = 100 kg·m²/(s³·A)
- 10 N / 5 Pa = (10 N)/(5 Pa) = 2 m²
- 5 m/s to s/m = 0.2 s/m
- 500 € - 20% to $ ≈ $451.04
- 500 megabit/s × 2 h to b?byte ≈ 419.09516 gibibytes

c) Physical constants

- k_e / G × a_0 = (coulombs_constant / newtonian_constant) × bohr_radius ≈ 7.126e9 kg·H·m^-1
- h / (λ_C × c) = planck / (compton_wavelength × speed_of_light) ≈ 9.1093837e-31 kg
- 5 ns × rydberg to c ≈ 6.0793194E-8c
- atom(Hg; weight) + atom(C; weight) × 4 to g ≈ 4.129e-22 g
- (G × planet(earth; mass) × planet(mars; mass))/(54.6e6 km)^2 ≈ 8.58e16 N (gravitational attraction between earth and mars)

d) Uncertainty and interval arithmetic

- `sin(5±0.2)^2/2±0.3 ≈ 0.460±0.088 (0.46±0.12)`
- `(2±0.02 J)/(523±5 W) ≈ 3.824±0.053 ms (3.825±0.075 ms)`
- `interval(-2; 5)^2 ≈ intervall(-8.2500000; 12.750000) (intervall(0; 25))`

e) Algebra

- `(5x^2 + 2)/(x - 3) = 5x + 15 + 47/(x - 3)`
- `(\a + \b)(\a - \b) = ("a" + "b")("a" - "b") = 'a'^2 - 'b'^2`
- `(x + 2)(x - 3)^3 = x^4 - 7x^3 + 9x^2 + 27x - 54`
- `factorize x^4 - 7x^3 + 9x^2 + 27x - 54 = x^4 - 7x^3 + 9x^2 + 27x - 54 to factors = (x + 2)(x - 3)^3`
- `cos(x)+3y^2 where x=pi and y=2 = 11`
- `gcd(25x; 5x^2) = 5x`
- `1/(x^2+2x-3) to partial fraction = 1/(4x - 4) - 1/(4x + 12)`
- `x+x^2+4 = 16  = x = 3 or x = -4`
- `x^2/(5 m) - hypot(x; 4 m) = 2 m where x>0  x ≈ 7.1340411 m`
- `cylinder(20cm; x) = 20L (calculates the height of a 20 L cylinder with radius of 20 cm)  = x = (1 / (2π)) m  = x ≈ 16 cm`
- `asin(sqrt(x)) = 0.2  = x = sin(0.2)^2  = x ≈ 0.039469503`
- `x^2 > 25x  = x > 25 or x < 0`
- `solve(x = y+ln(y); y) = lambertw(e^x)`
- `solve2(5x=2y^2; sqrt(y)=2; x; y) = 32/5`
- `multisolve([5x=2y+32, y=2z, z=2x]; [x, y, z]) = [-32/3 -128/3 -64/3]`
- `dsolve(diff(y; x) - 2y = 4x; 5) = 6e^(2x) - 2x - 1`

f) Calculus

- `diff(6x^2) = 12x`
- `diff(sinh(x^2)/(5x) + 3xy/sqrt(x)) = (2/5) × cosh(x^2) - sinh(x^2)/(5x^2) + (3y)/(2 × √(x))`
- `integrate(6x^2) = 2x^3 + C`
- `integrate(6x^2; 1; 5) = 248`
- `integrate(sinh(x^2)/(5x) + 3xy/sqrt(x)) = 2x × √(x) × y + Shi(x^2) / 10 + C`
- `integrate(sinh(x^2)/(5x) + 3xy/sqrt(x); 1; 2) ≈ 3.6568542y + 0.87600760`
- `limit(ln(1 + 4x)/(3^x - 1); 0) = 4 / ln(3)`

g) Matrices and vectors

- `[1, 2, 3; 4, 5, 6] = ((1; 2; 3); (4; 5; 6)) = [1 2 3; 4 5 6] (2×3 matrix)`
- `1...5 = (1:5) = (1:1:5) = [1 2 3 4 5]`
- `(1; 2; 3) × 2 - 2 = [(1 × 2 - 2), (2 × 2 - 2), (3 × 2 - 2)] = [0 2 4]`
- `[1 2 3].[4 5 6] = dot([1 2 3]; [4 5 6]) = 32 (dot product)`
- `cross([1 2 3]; [4 5 6]) = [-3 6 -3] (cross product)`
- `[1 2 3; 4 5 6].×[7 8 9; 10 11 12] = hadamard([1 2 3; 4 5 6]; [7 8 9; 10 11 12]) = [7 16 27; 40 55 72] (hadamard product)`
- `[1 2 3; 4 5 6] × [7 8; 9 10; 11 12] = [58 64; 139 154] (matrix multiplication)`
- `[1 2; 3 4]^-1 = inverse([1 2; 3 4]) = [-2 1; 1.5 -0.5]`

h) Statistics

- mean(5; 6; 4; 2; 3; 7) = 4.5
- stdev(5; 6; 4; 2; 3; 7) ≈ 1.87
- quartile([5 6 4 2 3 7]; 1) = percentile((5; 6; 4; 2; 3; 7); 25) ≈ 2.9166667
- normdist(7; 5) ≈ 0.053990967
- spearman(column(load(test.csv); 1); column(load(test.csv); 2)) ≈ -0.33737388 (depends on the data in the CSV file)

i) Time and date

- 10:31 + 8:30 to time = 19:01
- 10h 31min + 8h 30min to time = 19:01
- now to utc = "2020-07-10T07:50:40Z"
- "2020-07-10T07:50CET" to utc+8 = "2020-07-10T14:50:00+08:00"
- "2020-05-20" + 523d = addDays(2020-05-20; 523) = "2021-10-25"
- today - 5 days = "2020-07-05"
- "2020-10-05" - today = days(today; 2020-10-05) = 87 d
- timestamp(2020-05-20) = 1 589 925 600
- stamptodate(1 589 925 600) = "2020-05-20T00:00:00"
- "2020-05-20" to calendars (returns date in Hebrew, Islamic, Persian, Indian, Chinese, Julian, Coptic, and Ethiopian calendars)

j) Number bases

- 52 to bin = 0011 0100
- 52 to bin16 = 0000 0000 0011 0100
- 52 to oct = 064
- 52 to hex = 0x34
- 0x34 = hex(34) = base(34; 16) = 52
- 523<<2&250 to bin = 0010 1000
- 52.345 to float ≈ 0100 0010 0101 0001 0110 0001 0100 1000
- float(01000010010100010110000101001000) = 1715241/32768 ≈ 52.345001
- floatError(52.345) ≈ 1.2207031e-6
- 52.34 to sexa = 52°20′24″
- 1978 to roman = MCMLXXVIII
- 52 to base 32 = 1K
- sqrt(32) to base sqrt(2) ≈ 100000
- 0xD8 to unicode = Ø
- code(Ø) to hex = 0xD8