CALCULATIONS AND EXPRESSIONS

Cabri allows you to perform calculations using either the **Calculator** tool or by creating, editing and evaluating mathematical expressions.

1. THE CALCULATOR TOOLS

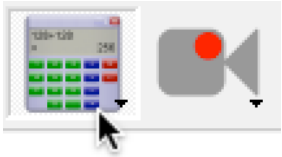
Cabri has two types of calculator. The default primary calculator will perform calculations involving +, -, ×, ÷ only, while the secondary calculator has a wider range of available operations: powers, trigonometric functions and squares and square roots.

To change between the two calculators, select the menu option **File – Activity Book Options** and then choose either Primary or Secondary level.

Each of the calculators has two possible modes, which can be toggled between in Author, Teacher and Student mode. The Author, however, can set the calculator to a particular mode and then, by deselecting the menu option **Page – Settable mode in calculator**, fix the calculator in this mode for Teachers and Students.

1.1 Numerical Mode

Select the **Calculator** tool.



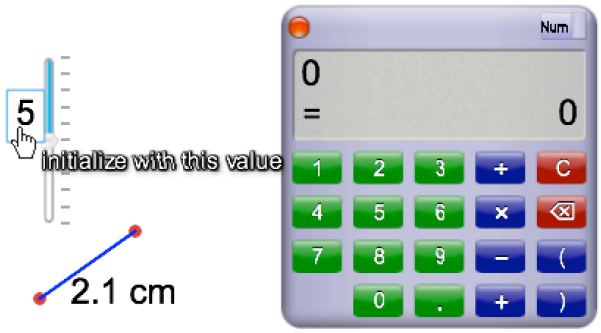
A calculator will appear above the page. The calculator may be dragged to change its location.

Primary calculator

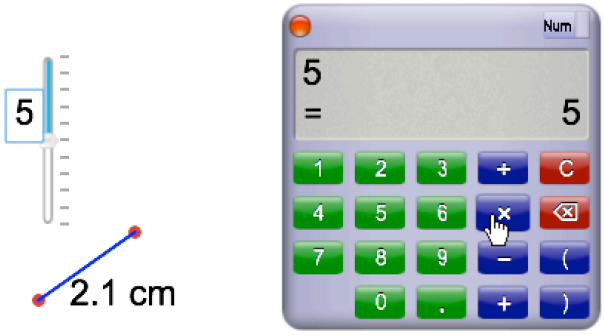
Secondary calculator

Note that the calculators are both in Numerical mode.

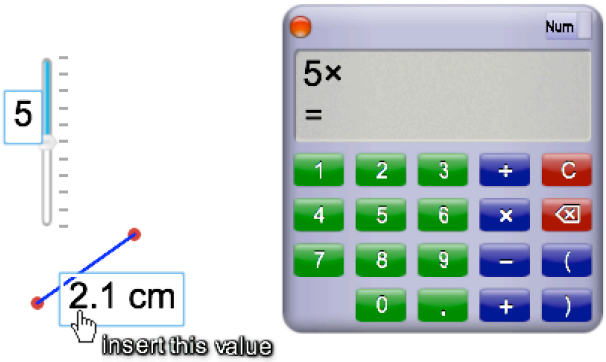
Put the cursor over a number you wish to include (in this case you could choose the slider itself rather than its associated number).



Click on the number (or slider) and then select the operation you wish to perform by clicking on the button or typing *.

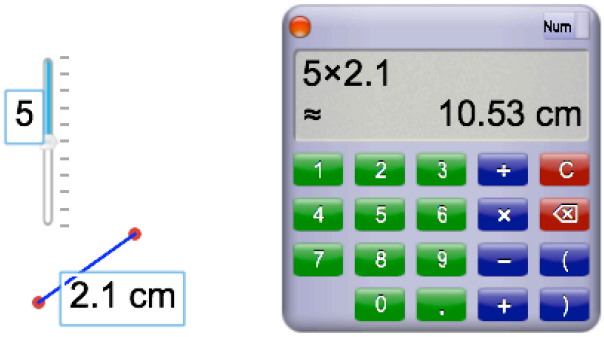


Now select a further number.

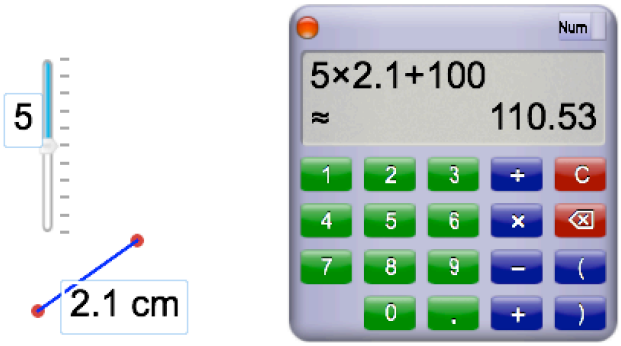


The calculation will be performed.

The result is displayed with two decimal place accuracy.



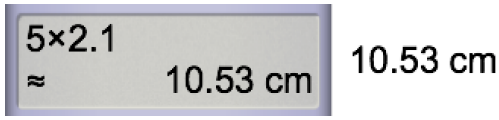
Add a further operation and type in a number directly. The new calculation will be performed.



If you want the result of the calculation on the page, put the cursor over the answer.

Drag it to the view and then release. The result may also be dragged and dropped directly into text, input boxes, table cells, and MCQs.

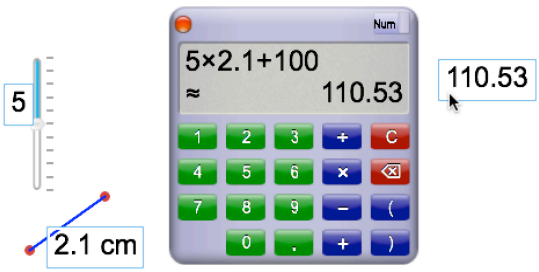
Note that units do not appear in the calculation, but will appear in the result and on the page.



To show further decimal places in the measurement or the result, select the number and press `ctrl +` (`ctrl –` for fewer)

With the secondary calculator, you can also change the accuracy displayed on the calculator screen.

Double-clicking on a calculated result (with the calculator CLOSED) will open the calculator with the calculation shown: it may then be edited.



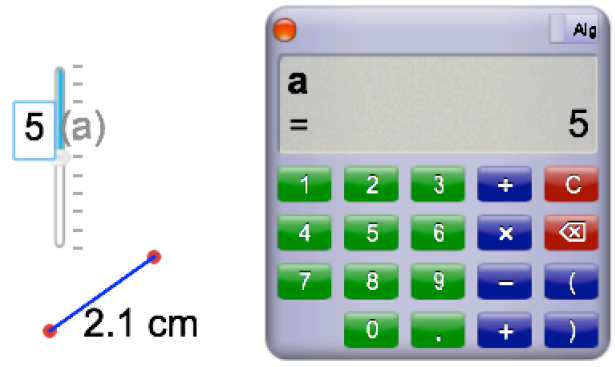
1.2 Algebraic Mode

We will repeat this calculation in algebraic mode. The only difference is in the way the calculation is displayed.

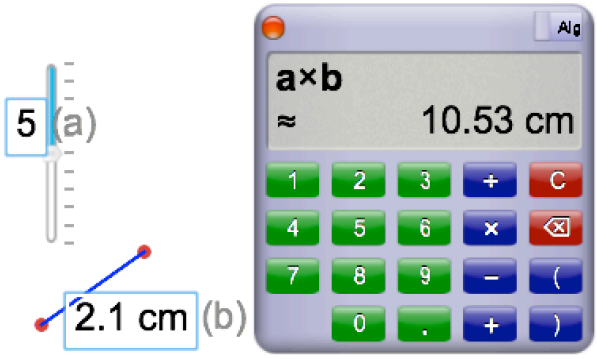
When the calculator is open, click on the Num at its top right to change to algebraic mode.

The calculator is now in algebraic mode. It will remain in this mode for all subsequent uses until returned to numerical mode.

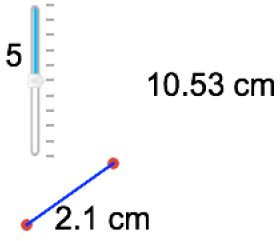
This time, when the 5 is selected, an **a** appears in the calculator and also beside the 5 on the screen.



A **b** appears beside the 2.1 cm and also in the calculator when the length is selected.

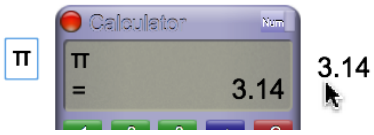


Once the result is entered on the screen and the calculator is closed, the variable names beside the numbers disappear.

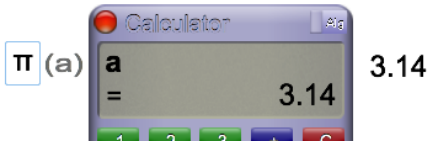


1.3 Calculations with Pi

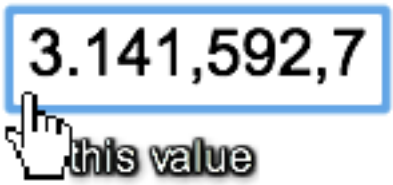
The number π may be used in the calculator. When selected in numerical mode it appears as a symbol until evaluated. It appears as a specific numerical value when evaluated and dropped in the view.



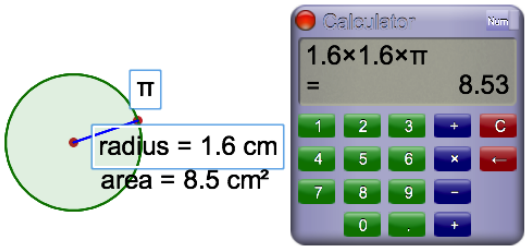
In algebraic mode, π is represented by a variable name until evaluated.



You can show π to up to 7 decimal places by pressing `ctrl +`.

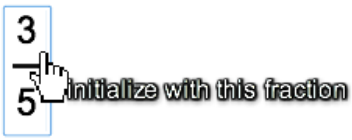


Here is an example using π to calculate the area of a circle.

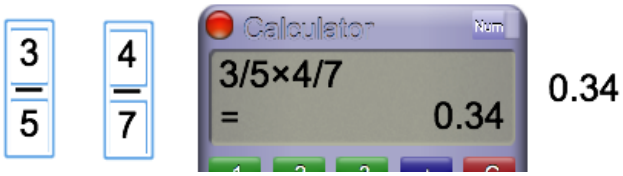


1.4 Calculations with fractions

An entire fraction, or just its numerator or denominator may be entered in the calculator.



If the entire fraction is entered, it is shown as a fraction, but any results from the calculation will be in the form of decimal fractions.



To perform arithmetic with fractions in the form of fractions, it is

necessary to use expressions.

1.5 Calculations involving Booleans

Booleans may also be used as input to the calculator: the operator “+” acts as OR and the operator “x” acts as AND.

2. EXPRESSIONS

More complicated calculations require the use of expressions. See Cabri Author file ***expressions*** page 1.

x	-3	-2	-1	0	1	2	3
(x - 1) * (x + 1)	8	3	0	-1	0	3	8

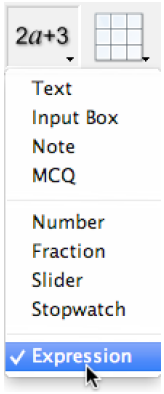
The expression (x – 1) * (x + 1) above has been evaluated for the values of x shown. If the expression is changed to e.g. (x – 2) * (x + 1) the results are updated instantly:

(x - 2) * (x + 1)	10	4	0	-2	-2	0	4

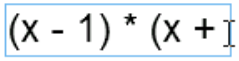
We will explain here how to create an expression and how to evaluate it. The mathematical operators, units, functions, and constants supported by the software are listed at the end of this document.

2.1 Creating an expression

Select the **Expression** tool and click in the working area to enter the expression editor.



Type in the expression that you want.



Note that a variable name can be a single letter (‘a’, ‘b’, etc), or a simple word such as “theCount”, “circle_radius” or “x0”.

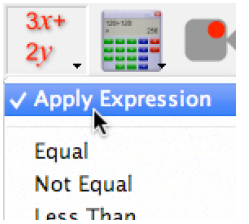
When you have finished, press `enter` or `esc`, or click somewhere in the view.

(x - 1) * (x + 1)

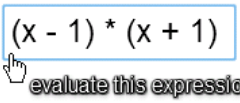
To later edit the expression, select the **Expression** tool, click on the expression and change it as required.

2.2 Evaluating an expression

Select the **Apply Expression** tool.



Click on the expression.



Now click on the number to be substituted for each variable, in the order in which the variables appear in the expression.

The result when the expression is evaluated will appear: click in the view to register and position it.

2.3 Expressions for arithmetic with fractions

We will create a series of expressions that will enable us to add two fractions and give the answer as another fraction. See Cabri Author file ***expressions*** page 2. Similar series of expressions may be calculated for other operations with fractions.

Start with two fractions.

$\frac{5}{6}$ $\frac{4}{9}$

Create an expression for the denominator of the sum (not necessarily in lowest terms).

den(a)×den(b) 54

Create and evaluate an expression to multiply the numerator of one fraction by the

num(a)×den(b) 45 24

denominator of the other.

Create and evaluate an expression to add these two numbers. The result will be the numerator of the sum (not necessarily in lowest terms).

a+b 69

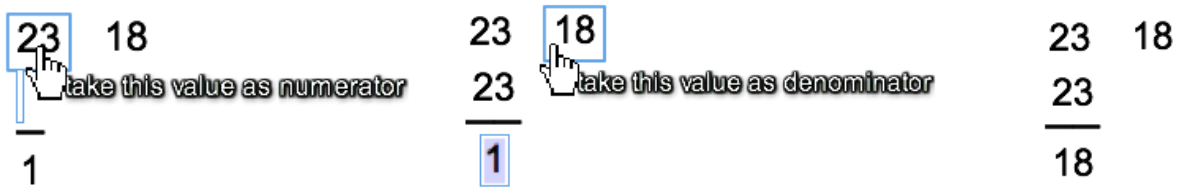
Now find the greatest common factor of the numerator and denominator.

pgcd(a;b) 3

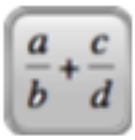
Divide the numerator and denominator by this number to get the numerator and denominator of the sum in its lowest terms.

a÷b 23 18

Create a new fraction with these values as numerator and denominator.



In a future release, it will be possible to turn this sequence of calculations into a macro by choosing the first two fractions as initial objects and the resulting fraction as final object.

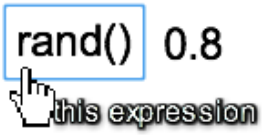


See the **MACROS** documentation for details.

2.4 Using expressions to generate random numbers

See Cabri Author file ***expressions*** page 3.

The expression **rand()** will create a random number in the interval [0,1) (i.e. including 0 but not including 1).



To change the value, double-click on the expression and then click elsewhere.

rand() 0.12

For more (or fewer) decimal places, press `ctrl +` (`ctrl -`).

rand() 0.353

Let's modify this expression so that it will give us a random integer between 1 and 10.

1+floor(10×rand()) 10

It is possible to recalculate an expression using the action "Set Values" with the expression as parameter. See the **ACTIONS** documentation for details.

Let's create a button with this action. Click on the expression TWICE when selecting it.

Clicking on the button generates a different random number.

1+floor(10×rand()) 1

We can generalize this expression further. The expression to the left will generate random integers in the interval [**a**, **a**+**b**-1].

a+floor(b×rand())

Evaluate this expression once with particular values of **a** and **b** (in this case **a** = -2 and **b** = 5).

a+floor(b×rand()) 1
-2 5

Now create a similar button to reset the value of the expression. Each button click gives a different random integer in the interval [-2, 2].

a+floor(b×rand()) 2
-2 5

If you change the value of a or b, the interval for the random number will change accordingly.

a+floor(b×rand()) 14
10 5

2.5 Randomizing the Random numbers

Random numbers are set up in Cabri so that when a file is opened the first number generated is always the same (to enable teachers and students to see the same initial page). Further random numbers will be

different to enable different students to have different numbers, and for a different sequence to appear if an activity is redone by a student.

Occasionally it might be useful for the first number to be different each time. We can do this by generating a large number of random numbers for a variable amount of time, and only using the last number generated. See page 4 in the ***expressions*** file.

Use the **Circle** tool to create a circle between two points.

Use the **Hide/Show** tool to hide the radius point.

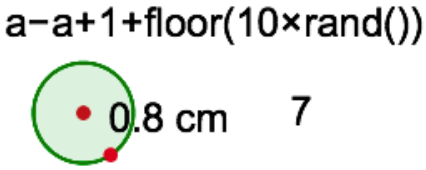
Use the **Point** tool to create a new point on the circle and use the **Distance** tool to find the distance between this point and the center of the circle.

Use the **Expression** tool to create the expression to the left.

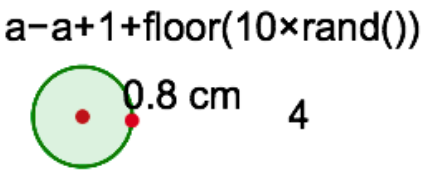
a−a+1+floor(10×rand())

This will give a random number in the interval [1,10], as the value of this expression is unaffected by the value of **a.**

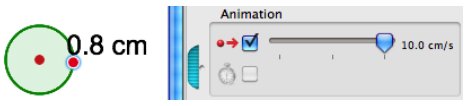
Now evaluate the expression with this measurement for **a**.



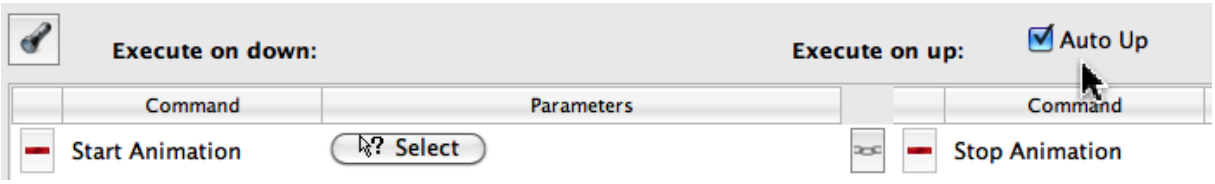
Changing the position of the point on the circle will cause the distance to be recalculated (even though it doesn't change) and hence the expression is reevaluated, which changes the random number generated.



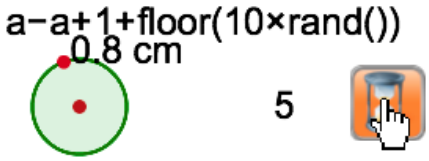
Now animate the point on the circle with a fast speed. See the **ANIMATION** documentation for details.



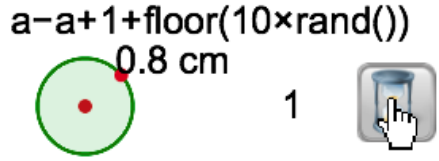
Create a toggle button with the actions shown, and parameter the point. Make sure the **Auto Up** box is selected.



When this button is pressed down, the point moves and new random numbers are rapidly generated.



When the button is released, the point stops, and the last number generated remains.



The amount of time that the button is held down, which will always vary, determines the number that results.

Another option would be to start the animation with one action, such as entering a page, and stop the animation with another action, such as moving to a different page. See the **MULTIPLE CHOICE QUESTIONS** documentation for an example of the use of random numbers to generate a series of questions.

3. OPERATORS, UNITS, FUNCTIONS, CONSTANTS

3.1 List of supported operators

Symbol	*Description*
+	scalar addition or Boolean OR operator
-	scalar subtraction
*	scalar multiplication or Boolean AND operator
/	scalar divide
√	square root (*secondary calculator only*)
²	square (*secondary calculator only via* f^{-1} *button*)
^	scalar power or Boolean XOR operator *(secondary calculator and expressions only)*
#	integer Euclidean division (e.g. 13#5 = 2) *(expressions only)*

% modulus (e.g. 13%5 = 3) *(expressions only)*

Cabri follows the standard order of operations: brackets, then exponents (^), then multiplication or division (*, /, #, %) from left to right, then addition or subtraction (+, -) from left to right.
For example:
5 – 2 / 2 * 3 = 5 – 1 * 3 = 5 – 3 = 2 while (5 – 2) / (2 * 3) = 3 / 6 = 0.5

3.2 List of supported units

Symbol	*Description*	*Aliases*
cm	centimeter	
cm²	square centimeter	cm2
cm³	cubic centimeter	cm3
°	degree (angle)	deg
rad	radian (angle)	

3.3 List of supported mathematical functions and constants

Note: with the exceptions of *num, den, not* and *rand*, all functions require a real argument (*x*) or arguments (*x;y*), defined between brackets, and return a real value. *num* and *den* have different behaviours dependent on whether the argument is a fraction or a real number. The *not* function requires a Boolean argument and returns a Boolean value. The *rand* function has no argument.

Name	*Description*	*Aliases*
abs(*x*)	absolute value of *x*	Abs
round(*x*)	nearest integer to *x*	Round
floor(*x*)	floor of *x*	Floor
ceil(*x*)	ceil of *x*	Ceil
trunc(*x*)	integer part of *x*	Trunc
sign(*x*)	sign of *x* (-1, 0 or 1)	Sign
num(*f*)	numerator of fraction *f*	Num
num(*x*)	*x* if *x* not fraction	
den(*f*)	denominator of fraction *f*	Den
den(*x*)	1 if *x* not fraction	

not(*b*)	boolean negation of *b*	Not
rand()	random number in [0,1) *(may give 0, will not give 1)*	Rand
min(*x;y*)	minimum of *x* and *y*	Min
max(*x;y*)	maximum of *x* and *y*	Max
pgcd(*x;y*)	GCD of *x* and *y*	Pgcd
plcm(*x;y*)	LCM of *x* and *y*	Plcm
sqrt(*x*)	square root of *x*	Sqrt
sqr(*x*)	square of *x*	Sqr
exp(*x*)	exponent of *x*	Exp
ln(*x*)	natural logarithm of *x*	Ln
log(*x*)	base-10 logarithm of *x*	Log, lg, Lg
lb(*x*)	binary logarithm of *x*	Lb
cos(*x*)	cosine of *x* *(also on secondary calculator)*	Cos
sin(*x*)	sine of *x* *(also on secondary calculator)*	Sin
tan(*x*)	tangent of *x* *(also on secondary calculator)*	Tan
acos(*x*)	inv. cosine of *x* *(also on secondary calculator via* f^{-1} *button)*	ArcCos, arcos
asin(*x*)	inv. sine of *x* *(also on secondary calculator via* f^{-1} *button)*	ArcSin, arcsin
atan(*x*)	inv. tangent of *x* *(also on secondary calculator via* f^{-1} *button)*	ArcTan, arctan
cosh(*x*)	hyperbolic cosine of *x*	CosH, ch, Ch
sinh(*x*)	hyperbolic sine of *x*	SinH, sh, Sh
tanh(*x*)	hyperbolic tangent of *x*	TanH, th, Th
acosh(*x*)	inv. hyperbolic cosine of *x*	argch, ArgCh
asinh(*x*)	inv. hyperbolic sine of *x*	argsh, ArgSh
atanh(*x*)	inv. hyperbolic tangent of *x*	argth, ArgTh

Note: the trigonometric functions (*cos*, *sin*, *tan*) require a scalar angle as argument. If the argument is not explicitly defined in *degrees* (°), the software assumes that the argument is defined in *radians* (rad). For example, “cos(60)” or “cos(60 cm)” is equivalent to cos(60 rad) = -0.95, while cos(*x*) where *x* is an angle measured as 60°, gives cos(60°) = 0.5.

3.4 List of supported mathematical constants

pi	π number (3.1415…)	Pi, PI
true	Boolean true value	True, TRUE
false	Boolean false value	False, FALSE