How to Use DragginMath

This is a very brief overview of how to use DragginMath. There is more to know, but this will get you started.

Read *all* of a paragraph *before* you try to do anything described in that paragraph.

You will see ℹ️ in many places throughout DragginMath, each with its own unique information page. Be sure to tap ℹ️ and read these at some point (but *not now*), as this app has many features that hide in plain sight. If you are reading the main ℹ️ page, you may have to swipe up and down to see all topics. Tap OK at the bottom when you finish reading.

The ☰ button in the upper right corner raises a list of configuration switches that modify DragginMath's behavior to suit your needs. You may have to swipe up and down to see all switches. Each switch has its own unique ℹ️ page. Some of these issues are simple and obvious, while others are subtle and sophisticated. **The default switch settings are good to get you started.** Tap OK at the bottom when you finish setting switches.

Tap inside the text field at the top of the screen whenever you want to start a new math problem. This erases the screen and raises the screen keyboard. **Tap the Top Text field now.**

Some columns of keys swipe up and down to expose

more symbols. These *key columns* have colored *column backgrounds* to help you know they can be swiped. For example, the column containing = is one of these. If you don't see a symbol you want to use, swipe these columns up and down to find it. Tap ℹ️ *on the screen keyboard* to learn more about this and other keyboard features.

The symbols ＝ and =꞉ are *not* interchangeable. For now, ＝ is what you want.

↑ is the exponentiation (raise) operator. 2↑5 means "two to the fifth power".

DragginMath only accepts expressions that are correct and complete. If you tap a key whose symbol *cannot* be correct where you tried to put it, DragginMath rejects it immediately: the text field flashes red and there is a (((*bonk*))) sound. If you tap a key whose symbol is mathematically acceptable but that you typed by mistake, use backspace ⬅️ to fix it. **You can only add or remove symbols at the end of the line.** Inserting or deleting in the middle doesn't work here.

The keyboard won't go away until you tap endline ⏎. Look for the green ⏎ key in a lower corner of the keyboard. **If the expression in the Top Text is not mathematically complete, ⏎ is not accepted.** If you just want the keyboard to go away, use backspace ⬅️ to erase everything, then tap ⏎.

Use the keyboard to enter **2+3+4** and ⏎. Look for the diagram in the middle of the screen. This kind of diagram is

called an ***operator tree***. It means the same thing as the expression you entered in the Top Text, but it also shows how the symbols are connected in meaning. ***This kind of diagram is the reason this app exists.*** Whenever you change either the text or the tree, DragginMath changes the other to match it.

Drag your fingertip on whitespace in the main display area to move all operator trees on the screen. Double-tap on whitespace to send trees home to the upper left corner. Try it now.

Lay your fingertip on + at the top of the operator tree and leave it there. Look for the red circle around your fingertip. The circle means DragginMath knows you are touching *that node* in the operator tree. **Drag the *root node* of a tree to move the whole tree.** If there is only one tree on the screen, this is the same as dragging whitespace.

Lay your fingertip on **2** in the operator tree and leave it there. Look in the Top Text to see **2** highlighted there. **Whenever you touch a node in a tree, the corresponding symbol is highlighted in the Top Text.**

With your fingertip still on **2**, drag it to the *left* and then lift your finger. See **2** go back where it came from. You moved something, but DragginMath couldn't figure out what that movement means, so it set everything back the way it was before you moved it. This is typical DragginMath behavior. If you want to know if something works, *try it*. **If your movement doesn't mean anything to DragginMath, *nothing happens*.**

Put your fingertip back on **2**, then drag it to the *right* past **3**. The red circle turns purple when you do this. Now lift your finger. See that **2** and **3** have changed places. Put your fingertip on **4**, drag it to the *left* until you see that **3+2** and **4** have changed places, then lift your finger. This is the ***Commutative Property*** in action. You can drag the nodes back the other direction if it serves your purpose.

There are now two + symbols on the screen. Let's call them *upper* and *lower*. Drag the lower + *up onto* the upper +. The red circle turns blue when you drag up. When one node is directly over another, four things happen at the same time:

1) there is an audible (((*click*)));
2) the white background changes color slightly;
3) a black border appears around the node;
4) a target icon 🎯 appears in the upper left corner of the screen.

This means you are *on target*, and you can lift your finger to drop on that node. If you drag *off target*, four things happen at the same time:

1) there is a lower-pitched audible (((*click*)));
2) the screen background color changes back to white;
3) the black border around the target node disappears;
4) the target icon 🎯 disappears.

If you lift your finger and drop off target, nothing happens and everything goes back the way it was.

At first, this seems like a lot going on. After you have used DragginMath for only a little while, you will not have to pay deliberate attention to these signals: you will *just know* when you are on target.

Drag the lower + onto the upper + and drop it there. DragginMath knows this means you want to use the ***Associative Property***, so it rebuilds the tree and the text to show this result. Now there are still two + symbols. They make a *different structure*, but they *mean* the same thing. This ability to ***change structure without changing meaning*** is what algebra *is*, and this is what DragginMath *does*. If a change in structure would cause a change in meaning, DragginMath won't do it. The ***Associative Property*** and ***Commutative Property*** are only two of a large number of transformations DragginMath knows how to do.

Elementary algebra problems usually ask you to **solve for x**, so let's do that now. Tap the Top Text field again, then enter **3x+4=10** and ↵. Drag **4** up onto = and drop it there. See the result **3x=10−4**. What just happened has several different names, such as the ***Additive Property of Equality*** or ***Do the Same Thing to Both Sides*** or ***Move the Operand to the Other Side*** or ***Invert the Operator***. Now double-tap − to ***Evaluate***. See the result **3x=6**. Drag **3** up onto = and drop it there to see **x=6÷3**. Double-tap ÷ to see **x=2**.

This is how most elementary problems are solved in DragginMath: drag an operand (**4** and **3** in the previous example) up onto = to invert it to the other side, then double-tap an operator (− and ÷ in the previous example) to do some arithmetic. Repeat these steps until there is nothing more to do. To solve more complicated problems, you must understand and use other tools, also, such as the ***Commutative Property*** and ***Associative Property*** shown earlier.

DragginMath can help you solve problems *much* more

complicated than this. Read ℹ️ ***About Colors and Modes*** and ℹ️ ***About Writing Math*** found on the main ℹ️ page when you are ready to learn more. The Guided Tour on the website is a more thorough description of what DragginMath can do for you.

Be bold: try things! You can't hurt anything. If the result is not what you want, or if you didn't understand it, or if you just want to see it happen again, **use undo ↩️ to go back.** You can undo ↩️ and redo ↪️ through all steps of the current problem. Tap ⬆️ to review your recent problems and the steps you took to solve them. Double-tap any problem or step to reload it, or tap OK at the bottom when you finish reading.

Here is something to know about and decide soon: The ***Solve Part*** ◉ ***Full*** switch under the ☰ button controls the amount of detail shown during inversion. By choosing ***Solve Part***, beginners see and participate in all of the inversion process. Users who already understand the basics of algebra prefer the default ***Solve Full*** setting in which this cleanup work is done automatically. The example **3x+4=10** described here was performed under ***Solve Full***. Learn more about this important choice by reading ℹ️ at the ***Solve Part*** ◉ ***Full*** switch.