



Cozmo Code Lab Reminder

By Anki Team

Table des matières

CUBE	3
DRIVE	3
ACTIONS	4
ANIMATIONS	4
EVENTS	5
CONTROL	6
SENSORS	7
DISPLAY	9
OPERATORS	10
DATA	11
LIST	11

CUBE

The following is a full glossary for blocks included in Code Lab's Constructor Mode. The glossary is also accessible inside of Code Lab with images of the blocks so you'll never be without it.

Please post any questions / feedback about specific blocks in the forum¹.



Cube 1



Cube 2



Cube 3

Reference image for Cube markers.

DRIVE

This category of blocks control Cozmo's locomotion. An important detail to make note of is that certain blocks use Cozmo's path planner, while others do not. If a block does use the path planner, Cozmo will drive around / avoid recognized objects—like his Cubes—that are found along the drive path.

- **drive [X] mm at [Y] mm/s** — Drive a distance of [X] millimeters at a speed of [Y] millimeters per second. Use a negative [X] distance to drive backwards. Max [Y] speed: 220 mm/s.
- **turn [X] ° at [Y] °/s** — Turn [X] degrees at a speed of [Y] degrees per second. A positive [X] angle turns him right, and a negative turn him left. Max [Y] speed : 200 °/s.
- **drive left wheel [X] mm/s, right [Y] mm/s** — Drive the left wheel at [X] millimeters per second and the right wheel at [Y] millimeters per second. Use negative [X] / [Y] speed to rotate wheels counter-clockwise. Max speed : [X] / [Y] 220 mm/s.
- **stop [wheels / head / lift / all]** — Stop any current movement on the selected motor(s).
- **dock with Cube [#]** — Make Cozmo drive to a specific Cube and dock with it, which means he will get in position to be able to pick up the Cube but without raising his lift.
 - *Tip: Cozmo must see the specific Cube in order to dock with it, so it is often useful to do a check with the "is Cube [#] visible" sensor.*

¹ <https://forums.anki.com/>

- **navigate [X] forward, [Y] to side, [Z] °** — Navigate Cozmo to a position relative to his current position: [X] millimeters to the front, [Y] millimeters to the side, and angled at [Z] degrees. A negative [X] forward value moves him backward. A positive [Y] side value moves him right, while a negative move him left. A positive [Z] angle turns him right, and a negative turn him left. Cozmo will use his path planner to try to avoid obstacles.
- **navigate in world to [X] x, [Y] y, [Z] °** — ADVANCED CODE BLOCK. Navigate Cozmo to a coordinate within his world map: [X] millimeters on the x-axis, [Y] millimeters on the y-axis, and facing a [Z] degree angle. Cozmo will use his path planner to try to avoid obstacles.
 - *Tip: This can be used to precisely map where you would like Cozmo to go. Note that because the origin point of Cozmo's internal map is something he controls (and not you), this code block is often most useful once you've calculated the relative distance between points you'd like him to navigate.*

ACTIONS

Action blocks control Cozmo's lift movement, head movement, cube lights, and the lights found on his back. While there are a total of five lights on his back—three stacked vertically, with an additional one on each side of the stack—only the center column is controllable. Selecting the color black for a light turns it off.

- **say [text]** — Make Cozmo speak the specified [text].
- **move lift [X] °/s** — Move the lift at a speed to [X] degrees per second. It will keep moving until stopped, or until min/max height reached. Note: the lift uses an angular speed because the its arm is technically rotating. Max [X] speed : 200 °/s.
- **move lift to [X] % at [Y] °/s** — Move the lift to [X] percent of max (0% = bottom, 100% = top) at a speed to [Y] degrees per second. **Note:** the lift uses an angular speed because the its arm is technically rotating. Max [Y] speed : 200 °/s.
- **move head to [X]° at [Y] °/s** — Move the head to [X] degrees (-25° = max down, 0° = level, 45° = max up) at a speed of [Y] degrees per second. Max [Y] speed : 150 °/s.
- **set backpack light [color]** — Set Cozmo's backpack lights to the specified color. Use black to turn off the lights.
- **on Cube [#] set [all lights / light #] to [color]** — Set the specified edge [1, 2, 3, 4 or all 4 at once] of the specified Cube [#] to [color]. Use black to turn off the lights.
- **on Cube [#] [spin / blink] lights in [color]** — On Cube [1, 2, or 3], [spin or blink] lights in [color]. The color black indicates "lights off".


ANIMATIONS



A full Cozmo animation generally consists of a combination of physical movement, sound, and a face animation. The animation block allows you to not only trigger a full animation, but also

isolate parts of Cozmo's physical movement. This is useful when you only need to utilize a portion of a full animation.

- **play [animation name] animation** — Play the specified animation on Cozmo. If you would like to lock his wheels, lift, or head, use the enable/disable animation blocks.
- **enable: [wheels / head / lift] in animations** — If you have previously disabled Cozmo's wheels, head, or lift during animations, use this block to turn them on again. Cozmo's wheels, head, and lift are enabled by default."
- **disable: [wheels / head / lift] in animations** — Disable Cozmo's wheels, head, or lift during all animations from now on, until they are enabled.
 - *Tip: This block can be useful if your project depends on Cozmo being aligned in a certain way. For example, if you want Cozmo to play a happy animation, but it's important he stays aligned to tap on a Cube, you can disable his wheels.*
- **play [sound name] sound** — Play the [sound name] sound.
- **play [sound name] sound and wait** — Play the [sound name] sound and wait for it to complete.
- **stop [sound name] sound** — Stop the [sound name] sound.
 - *Tip: To turn off Code Lab's default background music, stop "background" sound.*
- **advanced: play [text] SDK animation** — ADVANCED CODE BLOCK. The Cozmo SDK provides access to hundreds more of Cozmo's animations. The animation's id string can be played using this block.
 - To see a full list of animation id strings, you'll need to install the Cozmo SDK and run either the [remote_control.py](#)² program or [GrinningHermit's Cozmo Explorer Tool](#)³. *Note that we recommend using animation groups over animation strings because the latter can and do change over time, meaning your program may not work in the future without changes.*
- **advanced: play [text] SDK animation group** — ADVANCED CODE BLOCK. The Cozmo SDK provides access to hundreds more of Cozmo's animations. The animation group's id string can be played using this block.
 - To see a full list of the animation group ids (i.e., animation triggers), check the SDK documentation⁴.

EVENTS

The following are what are referred to as "hat blocks," and have the  shape. These are used to start scripts and will always be at the top of a block stack, much like a hat is found on the top of one's head.

- **when  clicked** — This block triggers when the green flag (start) button is tapped. Note that tapping the green flag button does two things: 1) Triggers all "when  clicked" blocks, and 2) Makes the other Event category blocks able to be triggered. For example, as you're working on your project and the green flag button isn't active, if Cozmo sees your face, a "when face is seen" block WON'T be triggered. Once the

² <https://www.youtube.com/watch?v=0gyPdws-fl4>

³ <https://forums.anki.com/t/cozmo-explorer-tool-v0-5/5622>

⁴ <http://cozmosdk.anki.com/docs/generated/cozmo.anim.html#cozmo.anim.Triggers>

green flag button is tapped and is active, if Cozmo sees your face, a “when face is seen” block WILL be triggered.

- **when Cube [#] tapped** — Triggers this code block and runs the code blocks underneath it every time the specified Cube [#] is tapped.
- **when Cube [#] moved** — Triggers this code block and runs the code blocks underneath it every time the specified Cube [#] is moved.
 - *Tip: Moving a Cube means placing it in a slightly different position or location. Tapping a Cube means giving it a deliberate, quick tap.*
- **when face seen** — Triggers this code block and runs the code blocks underneath it every time Cozmo sees a face. Note that the green flag (start) button must be active for this block to trigger.
- **when happy face seen** — Triggers this code block and runs the code blocks underneath it every time Cozmo sees a happy face. Note that the green flag (start) button must be active for this block to trigger.
 - *Tip: Smile big!*
- **when sad face seen** — Triggers this code block and runs the code blocks underneath it every time Cozmo sees a sad face. Note that the green flag (start) button must be active for this block to trigger.
 - *Tip: Make a particularly upset frown.*
- **when [message#] received** — Triggers this code block and runs the code blocks underneath it whenever another code block broadcasts the corresponding message.
 - *Tip: For experienced coders, think of messages like functions.*
- **broadcast [message#]** — Broadcasts (sends) a message to trigger the “when [message#] received” block.
 - *Tip: After broadcasting a message, the blocks under this block will continue to execute. Therefore, both this original stack of blocks and the blocks underneath the “when [message#] received” block will run in parallel. For experienced coders, think of this like starting a new thread in most programming languages.*
- **broadcast [message#] and wait to complete** — Broadcasts (sends) a message to trigger the “when [message#] received” block, and then waits for the blocks under the “when [message#] received” to fully execute before continuing on.
 - *Tip: For experienced coders, think of this like a function call in most programming languages.*

CONTROL

As the name suggests, control blocks are used to control scripts and may take the form of a C block (for loops), hat block (for starting scripts), stack block (general block shape placed above / below others), or cap block (for stopping / cancelling a script).

- **wait [X] secs** — Wait [X] seconds. Troubleshooting: Does a project end before you’re expecting it to? If no code blocks are executing, a project will automatically end after a few seconds. Imagine you’re waiting for the player to tap a Cube, and you want them to be able to have more than a few seconds before the project stops running. Adding a “wait” block can make sure your project doesn’t end before you want it to.
- **repeat [X]** — Repeat the enclosed code [X] times.

- **forever** — Repeat the enclosed code forever until program is stopped.
Troubleshooting: Seeing your code run slowly? Try adding a short “wait” block to your forever loop. Sometimes the code can use a moment to catch its breath.
- **if (condition) then [A]** — If the (condition) is true, then do the blocks in [A].
- **if (condition) then [A] else [B]** — If the (condition) is true, then do the blocks in [A], otherwise do the blocks in [B].
- **wait until (condition)** — Wait until the (condition) evaluates to true.
- **repeat until (condition)** — Repeat forever until the (condition) evaluates to true.
 - **Troubleshooting:** Seeing your code behaving slowly? Try adding a short “wait” block to your forever loop. Sometimes the code can use a moment to catch its breath.
- **stop [all / this script]** — Use this block to either stop [all] code blocks in the entire project, or to stop [this script] and end all code blocks in the current stack of blocks.
- **stop [what Cozmo’s doing]** — Stop what Cozmo’s doing: [driving / moving head / moving lift / animations / saying text / everything]
- **wait for Cozmo to finish [what Cozmo’s doing]** — Wait for Cozmo to complete [driving / moving head / moving lift / animations / saying text / everything] before moving onto the next block.
- **enable: always wait for Cozmo to finish** — Enable automatically waiting for Cozmo to complete what he’s doing before moving onto the next block.
 - **Note:** *This is the default behavior.*
- **disable: always wait for Cozmo to finish** — Disable automatically waiting for Cozmo to complete what he’s doing before moving onto the next block.

SENSORS

These are all inputs (“reporters” in Scratch terminology) that allow you to read current values from Cozmo and his world. You can view the current contents of any sensor by lightly tapping it in the workspace, but note that the displayed value does *not* update in real time. A new tap can be used to update the displayed value.

- **Cozmo lift height %** — The height of Cozmo’s lift as a % of its maximum (100%) height.
- **Cozmo head angle °** — Returns the angle of Cozmo’s head in degrees. (-25° = max down, 0° = level, 45° = max up)
- **Cozmo pitch °** — Returns the angle, in degrees, that Cozmo is tilted forwards / backwards. Forward values are negative, backward values are positive.
- **Cozmo roll °** — Returns the angle, in degrees, that Cozmo is tilted right / left. Right values are positive, left values are negative.
- **Cozmo yaw °** — Returns the angle that Cozmo is facing right / left. Right values are positive, left values are negative.
- **Cozmo position [X/Y/Z] in world** — ADVANCED CODE BLOCK. Returns the [X], [Y] or [Z] component of Cozmo’s world position, using Cozmo’s internal map.
 - **Tip:** This can be used with the “navigate in world to [X] x, [Y] y, [Z] °” block to precisely map where you would like Cozmo to go. Note that because the origin point of Cozmo’s internal map is something he controls (and not you), this code block is often most useful once you’ve calculated the relative distance between points you’d like him to navigate.

- **is face visible** — Returns “true” or “false” depending on if Cozmo currently sees a face.
- **face expression** — The current expression on a visible face. Returns a string of “happy”, “upset” and “unknown”.
- **face name** — Returns the name of the face (i.e. the name entered in Meet Cozmo). Returns a zero-length string if he sees a face he doesn’t recognize.
- **face position in camera [X/Y]** — If a face is visible, returns the [X] or [Y] component of the 2D position of the center of the face in Cozmo’s camera view. (Cozmo’s camera’s range : 0,0 to 320,240)
 - *Tip: Try using face position as an input. If you move to the left or right, how would you like Cozmo to react?*
- **face position in world [X/Y/Z]** — ADVANCED CODE BLOCK. If a face is visible, returns the [X], [Y], or [Z] (height) position in Cozmo’s world. Note that the origin point of Cozmo’s internal map is something he controls (and not you). This is useful for determining the relationship between objects in Cozmo’s world.
- **was Cube [#] tapped** — Returns “true” or “false” depending on if a specified Cube was tapped.
- **last tapped Cube** — Returns [1/2/3] corresponding to the last Cube tapped.
- **is Cube [#] visible** — Returns “true” or “false” depending on if a specified Cube is visible. Note that a Cube being visible means that Cozmo can clearly see its marker, so the entire Cube needs to be within his line of sight and not too close, and the lighting conditions must be not too bright or dark.
- **Cube [#] pitch ° in camera** — Returns the angle, in degrees, of the specified Cube’s tip forwards/backwards as seen in Cozmo’s camera. Note that this requires Cozmo to see the Cube.
- **Cube [#] roll ° in camera** — Returns the angle, in degrees, of the specified Cube’s tilt right/left as seen in Cozmo’s camera. Note that this requires Cozmo to see the Cube.
- **Cube [#] yaw ° in camera** — Returns the angle, in degrees, of the Cube facing right/left (as if flat on a surface) as seen in Cozmo’s camera. Note that this requires Cozmo to see the Cube.
- **Cube [#] position in camera [X/Y]** — If the specified Cube is visible, returns the [X] or [Y] component of the 2D position of the center of the Cube in Cozmo’s camera view. (Cozmo’s camera’s range: 0,0 to 320,240).
- **Cube [#] position in world [X/Y/Z]** — ADVANCED CODE BLOCK. If a specified Cube is visible, returns the [X], [Y], or [Z] (height) position in Cozmo’s world map. Note that the origin point of Cozmo’s internal map is something he controls (and not you). This is useful for determining the relationship between objects in Cozmo’s world.
- **device pitch °** — The angle, in degrees, that the mobile device is tilted forward / backwards.
- **device roll °** — The angle, in degrees, that the mobile device is tilted right / left.
- **device yaw °** — The angle, in degrees, that the mobile device is facing right / left (as if flat on a surface).
- **current [time]** — Returns current [year / month / date / day-of-week / hour / minute / second].

DISPLAY

These blocks work by drawing to a temporary image that is only displayed on Cozmo's face when the "display on Cozmo's face" block is used.

- **display on Cozmo's face** — Display the pending image on Cozmo's face. Note that the image will stay on Cozmo's face for 30 seconds unless interrupted (by an animation, the program ending, another image being displayed, etc.).
- **clear all pixels** — Clear all pixels on the current pending image for Cozmo's face. Note that this does not immediately turn off all pixels on Cozmo's face. To do that, use this code block, then "display on Cozmo's face".
- **draw [text] at [X], [Y]** — Draw the specified [text] at the given pixel position, using [X],[Y] coordinates. Cozmo's face screen is 128 pixels wide and 64 pixels tall, so the visible [X] range is 0-127 and the visible [Y] range is 0-63. Coordinate 0,0 starts at the top left corner of Cozmo's face screen.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*
- **set text scale to [X] %** — Set how large the text on Cozmo's face is to [X] percent. 100% is the default size.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*
- **set text alignment to [top / center / bottom] [left / center / right]** — Aligns the text on Cozmo's face. [top / center / bottom] places the text in that alignment relative to Cozmo's face. [left / center / right] determines the text's alignment relative to the text position you've set with the "draw text at" block. For example, to draw text coming from the right side of Cozmo's face, set text alignment to "right", and in the "draw text at" block set the [X] value to 127 pixels.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*
- **draw line from [X], [Y] to [X2], [Y2]** — Draw a line between the 2 pixel coordinates. Cozmo's face screen is 128 pixels wide and 64 pixels tall, so the visible [X] range is 0-127 and the visible [Y] range is 0-63.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*
- **draw rectangle from [X], [Y] to [X2], [Y2]** — Draw a rectangle between the 2 pixel coordinates. Cozmo's face screen is 128 pixels wide and 64 pixels tall, so the visible [X] range is 0-127 and the visible [Y] range is 0-63.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*
- **fill rectangle from [X], [Y] to [X2],[Y2]** — Draw a filled rectangle between the 2 pixel coordinates. Cozmo's face screen is 128 pixels wide and 64 pixels tall, so the visible [X] range is 0-127 and the visible [Y] range is 0-63.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*
- **draw circle at [X], [Y] with radius [Z]** — Draw a circle at the given [X],[Y] pixel coordinate with the given [Z] radius in pixels. Cozmo's face screen is 128 pixels wide and 64 pixels tall, so the visible [X] range is 0-127 and the visible [Y] range is 0-63.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*
- **fill circle at [X], [Y] with radius [Z]** — Draw a filled circle at the given [X],[Y] pixel coordinate with the given [Z] radius in pixels. Cozmo's face screen is 128 pixels wide and 64 pixels tall, so the visible [X] range is 0-127 and the visible [Y] range is 0-63.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*

- **set drawing mode to [draw/erase] pixels** — Sets the drawing mode for whether you are adding or erasing pixels.
 - *Tip: to see what you're drawing, use the "display on Cozmo's face" block.*

OPERATORS

Operator blocks allow you to script math equations and handle strings.

- **[X] + [Y]** — Returns the sum of [X] plus [Y].
- **[X] - [Y]** — Returns [X] minus [Y].
- **[X] * [Y]** — Returns [X] multiplied by [Y].
- **[X] / [Y]** — Returns [X] divided by [Y].
- **([X] < [Y])** — True if [X] is less than [Y].
- **([X] = [Y])** — True if [X] is equal to [Y].
- **([X] > [Y])** — True if [X] is greater than [Y].
- **([X] and [Y])** — True if both (X) and (Y) sub-conditions are true.
- **([X] or [Y])** — True if either the (X) or (Y) sub-conditions are true.
- **(not [Y])** — True if (Y) is false, false if (Y) is true.
- **pick random [X] to [Y]** — Returns a random integer (whole number) between [X] and [Y].
- **[X] mod [Y]** — The modulo division of [X] and [Y] (i.e. what's the remainder if you divide [X] by [Y]).
- **round [X]** — Round [X] to nearest integer value (whole number), up or down.
- **[operators] of [X]** — Standard math operators: think operators that would be found on a scientific calculator.
 - [abs] = absolute value of [X]
 - [floor] = floor of [X]
 - [ceiling] = ceiling of [X]
 - [sqrt] = square root of [X]
 - [sin] = sine of [X]
 - [cos] = cosine of [X]
 - [tan] = tangent of [X]
 - [asin] = inverse of sine of [X]
 - [acos] = inverse of cosine of [X]
 - [atan] = inverse of tangent of [X]
 - [ln] = natural logarithm of [X]
 - [log] = logarithm of [X]
 - [e^] = e to the power of [X]
 - [10^] = 10 to the power of [X]
- **Join [text1] and [text2]** — Concatenate (combine) 2 strings of text together. For example: "Cozmo" and "says" becomes "Cozmosays".
- **letter [#] of [text]** — Returns the [#] letter of the given string of text. For example the #1 letter of "Cozmo" is "C". **Advanced Coder Tip:** It's possible to creatively use strings and this block to make and navigate a list.
- **length of [text]** — The length of the given string of text.

- **[text1] contains [text2]** — True if the second string of text is contained within the first string of text. For example, it is true “coz” is contained inside of “cozmo”.

DATA

Use data blocks for variable handling. This block category is empty by default. You must first create a variable before the “set...” and “change...” blocks will appear.

- **[Create variable ...]** — Create a new variable with a specified name.
- **set [variable] to [X]** — Set the value of the variable to X.
- **change [variable] by [X]** — Add X to the value of the variable.

LIST