

# Guide to LEGO® Powered Up programming blocks

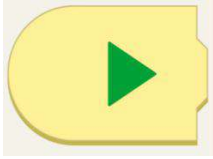
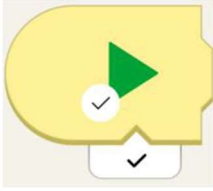
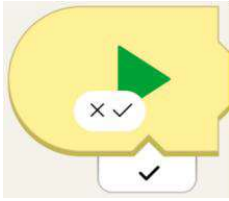
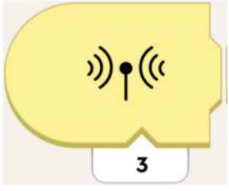
The LEGO® Powered Up app features a simple, easy-to-understand programming language that fans of all ages will enjoy learning and experimenting with. It's based around a core experience any LEGO fan will understand instinctively – stacking blocks next to one another and watching them snap into place.

Programming blocks come in a range of types and colors to tell you what type of block you're using.



Here are some examples of commonly used programming blocks. If you're not sure what a block does and it's not listed here, why not try it out in the app? Part of the fun of LEGO Powered Up is in the experimentation, play, and learning.

## Yellow Flow Blocks

**Yellow** blocks control the flow of your commands. You can use them to start a program, stop a program, pause a program or even loop commands.

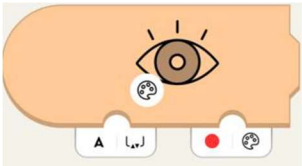
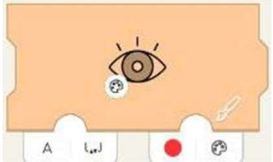
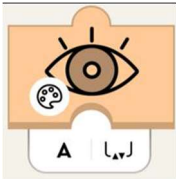
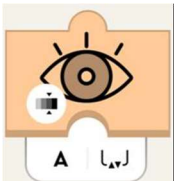
	This block starts the program.
	This block will start the program if a True condition is met.
	This block starts the program when the condition switches from False to True.
	This block starts the attached code sequence when the flag number is triggered.

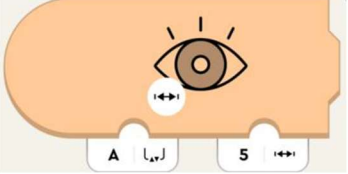
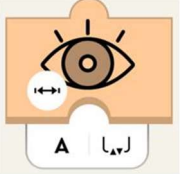
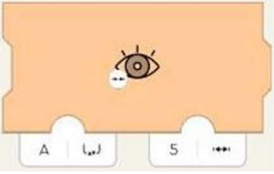
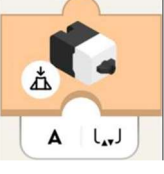

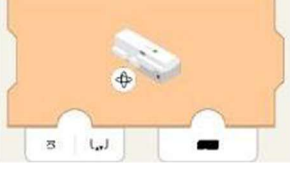

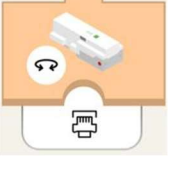
	<p>This block triggers a flag number.</p>
	<p>This hourglass block will make the program wait for a specified number of seconds.</p>
	<p>This hourglass block will wait for the condition to be True.</p>
	<p>This block will loop the program section for a set number of times.</p>
	<p>This block will loop the program while a condition is true.</p>
	<p>This block will loop the program forever!</p>
	<p>This block will run the top sequence if the condition is True. If the top sequence isn't True, it will run the bottom sequence.</p>
	<p>This block runs both sequences at the same time and waits for both sequences to finish before running the next block.</p>

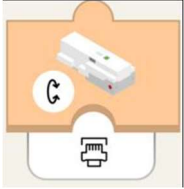
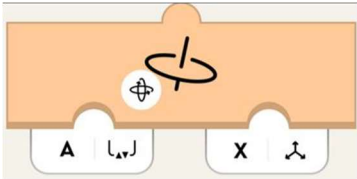
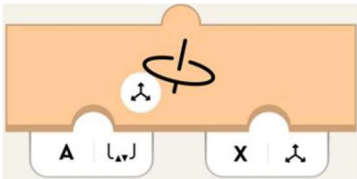
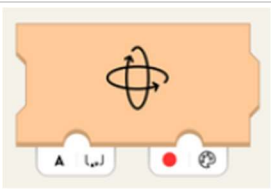

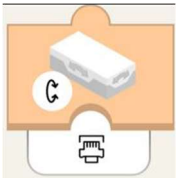
	<p>This block stops all other program sequences and continues running the attached code sequence.</p>
	<p>This block stops all sequences running on the current canvas. If it's used in the top program, this will end the program. If it's used in a Model Block and the Model Block is ended, the program containing the Model Block will continue running.</p>

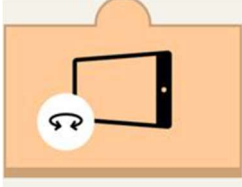
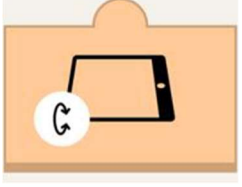
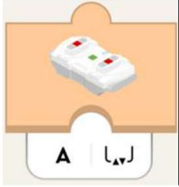
## Orange Sensor Blocks

**Orange** blocks work with the Color and Distance Sensor, the Move Hub's Tilt Sensor, remote control and WeDo 2.0 sensors. These blocks will trigger an action when a sensor on your model detects something.

	<p>Use the Color sensor to trigger an event when a specific color is sensed.</p>
	<p>This block tells the program to wait for the color sensor to detect the color chosen in the block.</p>
	<p>Get current color measured by the sensor.</p>
	<p>Get current ambient light level measured by the sensor.</p>

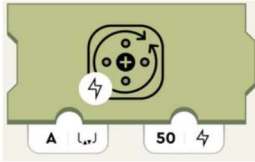
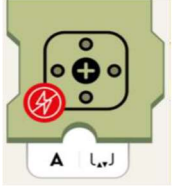
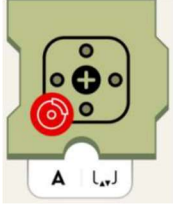
	<p>Triggers when the distance measured by the sensor is less than the specified distance.</p>
	<p>Get current distance measured by the sensor.</p>
	<p>Wait for the distance measured by the Sensor to be less than the specified distance.</p>
	<p>Get current force measured by the sensor and affect connected block functions.</p>
	<p>Use internal Gyroscope to trigger an event based on Orientation of the hub.</p>
	<p>Wait for the Hub orientation to be equal to Hub orientation.</p>
	<p>Get tilt orientation of the hub.</p>
	<p>Get X angle position of the hub.</p>

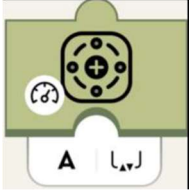


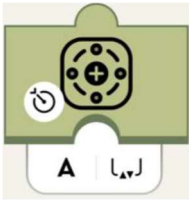


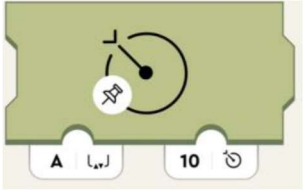
	<p>Get Y angle position of the hub.</p>
	<p>Get the angle of the specified axis on a tilt sensor.</p>
	<p>Get the specified axis on an internal accelerometer/tilt sensor of a hub.</p>
	<p>Sets the tilt orientation.</p>
	<p>Just for WeDo sensors - Measure distance</p>
	<p>Just for WeDo sensors. Get current WeDo sensor tilt angle on the Y axis (-90/90).</p>
	<p>Just for WeDo sensors. Get current WeDo sensor tilt angle on the X axis (-90/90).</p>

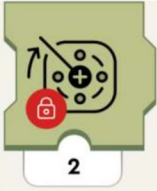
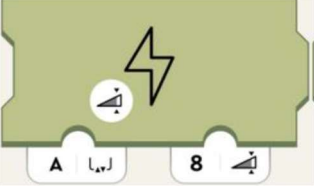
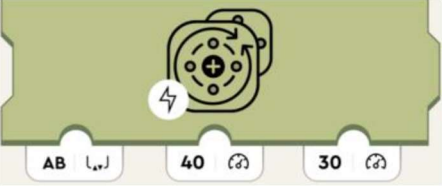
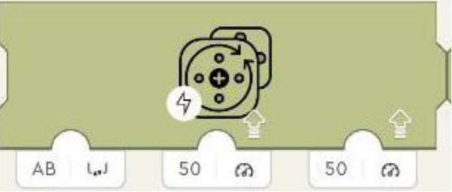
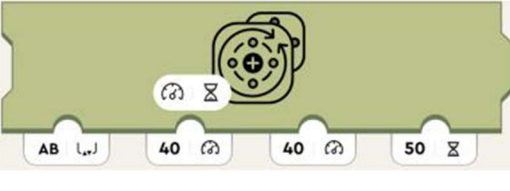
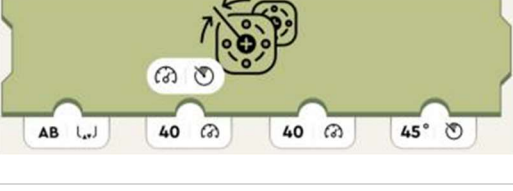
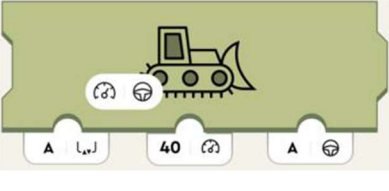
	<p>Uses the X axis tilt position of the smart device (left-right). 0 is the central position.</p>
	<p>Uses the Y axis tilt position of the smart device (forward to backward). 0 is around 30 degrees with the device screen facing you.</p>
	<p>Detects the button pressed on the A or B side of the connected remote.</p>

## Green Motor or Movement Blocks

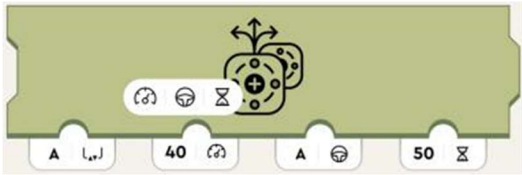
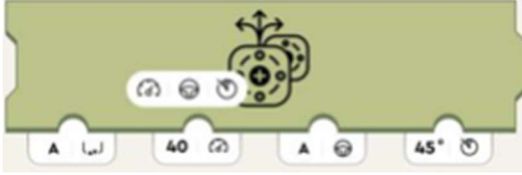
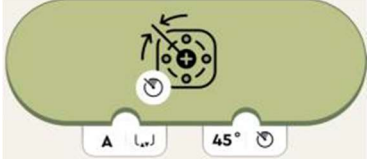
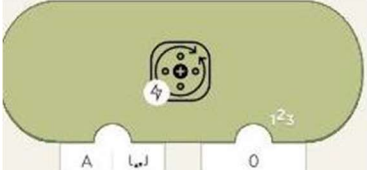
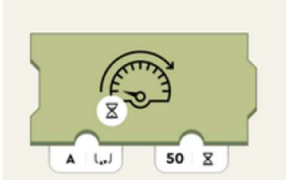
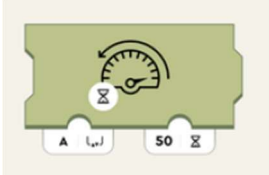
**Green** blocks get your model moving! You can differentiate between tacho motors (For controlling speed and position) and basic motors (To drive full speed).

	<p>Sets power to the motor. Positive numbers rotate clockwise, and negative numbers rotate counterclockwise.</p>
	<p>Removes power from the motor and will slowly stop the motors.</p>
	<p>Stops the motor.</p>

	<p>Reads the speed value of the tacho motor attached to the specified port.</p>
	<p>Makes a tacho motor run at a specified speed on a given port.</p>
	<p>Makes a tacho motor run on a specified port, at a specified speed for a specified duration (in seconds).</p>
	<p>Reads the position of the tacho motor. Position is relative. 0 is the position where the motor was turned on.</p>
	<p>Turns the tacho motor on a defined port at a defined speed with the defined degree.</p>
	<p>Turns the tacho motor on a defined port at a defined speed with the defined degree.</p>
	<p>Sets the current position of the tacho motor on a defined port to the defined value.</p>







	<p>Stops and holds the tacho motor on a defined port.</p>
	<p>This block is used to set the maximum power that a tacho motor should reach on a defined port.</p>
	<p>Sets the power of a pair of tacho motors on the defined ports, independently.</p>
	<p>Set the speed of a pair of tacho motors on the defined ports, independently.</p>
	<p>Sets the speed of a pair of tacho motors on the defined ports for duration (in seconds), independently.</p>
	<p>Sets the position of a pair of tacho motors on the defined ports with speed set independently.</p>
	<p>Sets the speed of a pair of tacho motors on the defined ports (AB or CD) with the steering direction defining a tank steering style behavior. 0 sets equal power to both outputs, 50 only powers one motor and 100 powers the motors in opposite directions. Positive-negative value for steering defines the direction.</p>

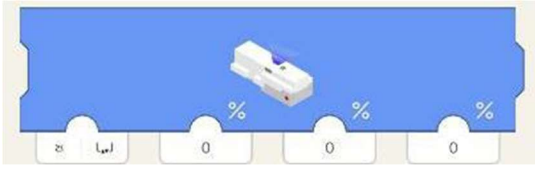








	<p>Sets the speed of a pair of tacho motors on the defined ports (AB or CD) with the steering direction defining a tank steering style behavior. The motors run for the defined duration (in seconds). 0 sets equal power to both outputs, 50 only powers one motor and 100 powers the motors in opposite directions. Positive-negative value for steering defines the direction.</p>
	<p>Sets the speed of a pair of tacho motors on the defined ports with the steering direction defining a tank steering style behavior. The motors run until the defined rotation degree is reached. 0 sets equal power to both outputs, 50 only powers one motor and 100 powers the motors in opposite directions. Positive negative value for steering defines the direction.</p>
	<p>Sets a tacho motor on the defined port to the given position in degrees.</p>
	<p>Sets power to the motor attached to the selected port and runs until the program runs without the need of a loop. Positive numbers rotate clockwise, and negative numbers rotate counterclockwise.</p>
	<p>Sets the acceleration duration (in seconds) of the tacho motor on a defined port.</p>
	<p>Sets the deceleration duration (in seconds) of the tacho motor on a defined port.</p>

## Purple Sound and Light Blocks

**Purple** blocks can play sounds through your device's speakers, change the color of the lights on the move hub, color and distance sensors. These blocks also control the lights on the LED-light.

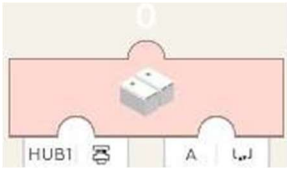
 <p>A [Port] 8</p>	<p>Sets the mode of the color and distance sensor connected to the defined port. 8 is the default value when connected, 7 is the infrared mode.</p>
 <p>A [Port] [Color]</p>	<p>Sets the color of the sensor LED.</p>
 <p>0 [Color]</p>	<p>Sets the hub LED to show color.</p>
 <p>A [Port] 50 %</p>	<p>Sets the light brightness on the connected to the specified port.</p>
 <p>A [Port] [Port] 0 % [Port] 0 % [Port] 0 % [Port] 0 %</p>	<p>Sets the light brightness on the connected to port (left parameter) but sets the four LED half-circles on the sensor individually.</p> <p>From left to right, they affect</p> <ul style="list-style-type: none"> <li>• Right "eye" upper arc</li> <li>• left "eye" upper arc</li> <li>• right "eye" <b>lower</b> arc</li> <li>• left "eye" lower arc</li> </ul>
 <p>0 0 0 0</p>	<p>Sends IR commands to the power function's IR receiver with the color and distance sensor connected to the defined port. The first input slot is the port where the sensor is attached. The second is the IR channel used from 0 to 3 matching IR port 1 to 4. The third defines multiple</p>

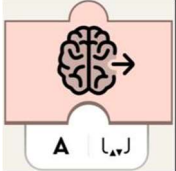
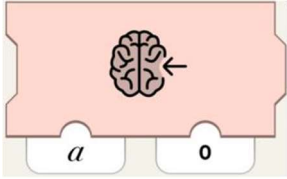
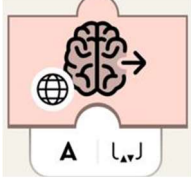
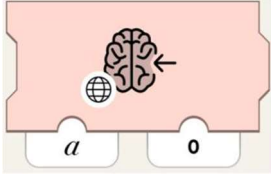
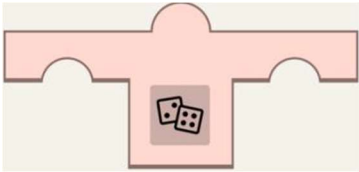
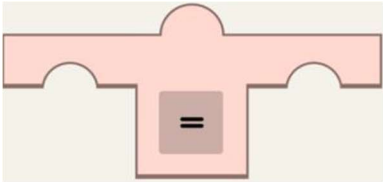
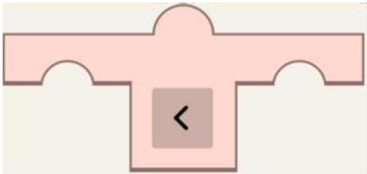
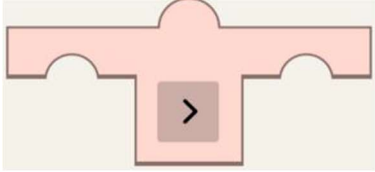
	<p>things but for basic use, set 4 to control the red port and 5 to control the blue port. The fourth sets the speed, 0 is float (no speed), 1 to 7 is increasing speed forward, 8 is stop, and 9 to F is decreasing speed backwards.</p>
	<p>Sets the hub LED to show color based on the RGB intensity defined (red, green, and blue value in %).</p>
	<p>Plays the selected sound from the house sound library.</p>
	<p>Plays the selected sound from the whistle sound library.</p>
	<p>Plays the selected sound from the train sound library.</p>
	<p>Plays the selected sound from the car sound library.</p>
	<p>Plays the selected sound from the plane sound library.</p>
	<p>Plays the selected sound from the dog sound library.</p>

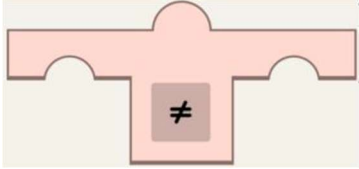
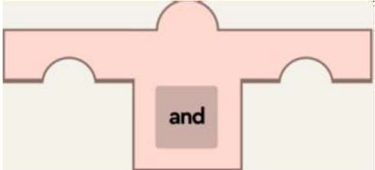
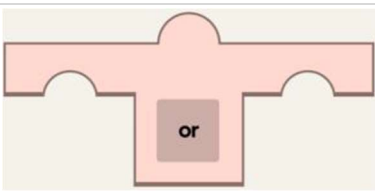
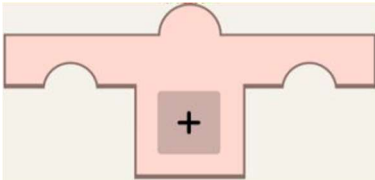
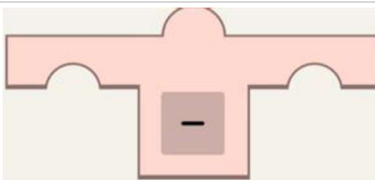
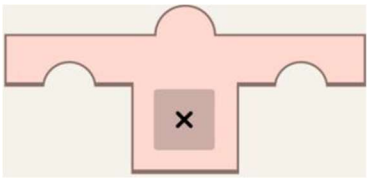
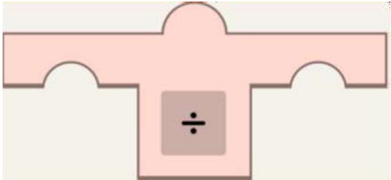
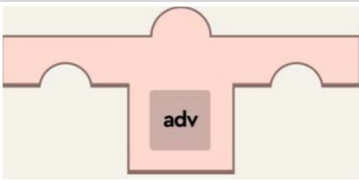
 <p>A blue sound block with a cat face icon, a speaker icon, and the number 2 in a white circle at the bottom.</p>	Plays the selected sound from the cat sound library.
 <p>A blue sound block with a fart cloud icon, a speaker icon, and the number 2 in a white circle at the bottom.</p>	Plays the selected sound from the fart sound library.
 <p>A blue sound block with a horn icon, a speaker icon, and the number 2 in a white circle at the bottom.</p>	Plays the selected sound from the horn sound library.
 <p>A blue sound block with a crowd of hands icon, a speaker icon, and the number 2 in a white circle at the bottom.</p>	Plays the selected sound from the crowd sound library.
 <p>A blue sound block with a robot head icon, a speaker icon, and the number 2 in a white circle at the bottom.</p>	Plays the selected sound from the robot sound library.
 <p>A blue sound block with a target icon, a speaker icon, and the number 2 in a white circle at the bottom.</p>	Plays the selected sound from the target sound library.

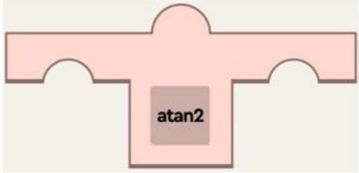
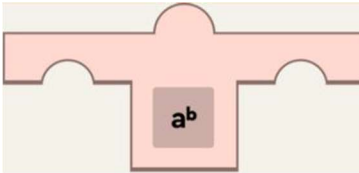
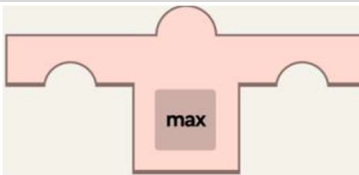
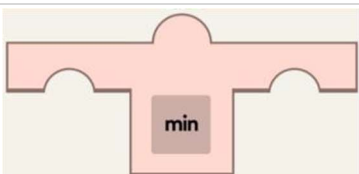
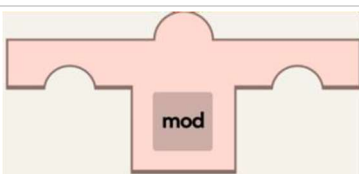
## Light Grey Math blocks

Light grey blocks allow for some complex programming using variables.

 <p>A light grey block with a USB icon in the center. Below the icon are two slots labeled 'HUB1' and 'A'. The number 0 is displayed above the block.</p>	Select a specific hub in the left slot and output the port used for that hub in right slot.
--	---


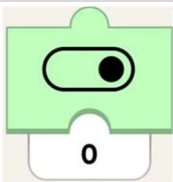
 <p>A Scratch 'Read Local Variable' block. It features a brain icon with an arrow pointing right. Below the icon, there is a label 'A' and a dropdown menu showing '1,2,3'.</p>	<p>Reads the number stored in the local variable.</p>
 <p>A Scratch 'Set Local Variable' block. It features a brain icon with an arrow pointing left. Below the icon, there are two input fields: the first contains the letter 'a' and the second contains the number '0'.</p>	<p>Update local variable to stored number.</p>
 <p>A Scratch 'Read Global Variable' block. It features a brain icon with an arrow pointing right and a globe icon to its left. Below the icon, there is a label 'A' and a dropdown menu showing '1,2,3'.</p>	<p>Reads the number stored in the global variable.</p>
 <p>A Scratch 'Set Global Variable' block. It features a brain icon with an arrow pointing left and a globe icon to its left. Below the icon, there are two input fields: the first contains the letter 'a' and the second contains the number '0'.</p>	<p>Update global variable to stored number.</p>
 <p>A Scratch 'Random Number' block. It features a central area with a dice icon, flanked by two input fields for numbers.</p>	<p>Outputs a random number between number and number (both numbers included in range).</p>
 <p>A Scratch comparison block with an equals sign '=' in the center. It has two input fields on either side for numbers.</p>	<p>Returns whether number is equal to number.</p>
 <p>A Scratch comparison block with a less-than sign '&lt;' in the center. It has two input fields on either side for numbers.</p>	<p>Returns whether number is less than number.</p>
 <p>A Scratch comparison block with a greater-than sign '&gt;' in the center. It has two input fields on either side for numbers.</p>	<p>Returns whether number is greater than number.</p>

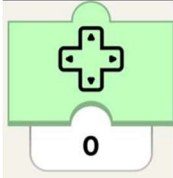
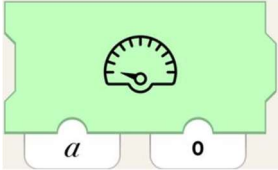
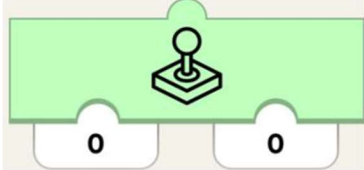
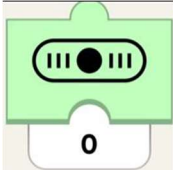
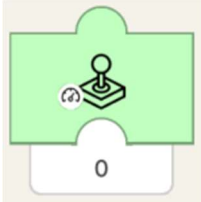

	Returns whether number is not equal to number.
	Returns true if both operands are true, and false otherwise Example: Returns true if both input values (left and right) are true, and false otherwise.
	Returns true if either operand is true, and false otherwise
	Returns the result of adding number to number.
	Returns the result of number subtracted with number.
	Returns the result of number multiplied with number.
	Return the result of number divided by number.
	Return an adorning with multiple math operations. Example: Returns a value from math operator (left slot) and numerical value (right slot).

	Returns a numeric value between $-\pi$ and $\pi$ representing the angle of an (left slot, right slot) point.
	Return the power of the left input powered with the right.
	Return the highest of the values.
	Return the lowest of the values.
	Divides left with right and gives the remainder from that operation.

## Teal Widget Blocks

Teal blocks let you control your model via widgets.

	Reports the state (true/false or 0/1) of the widget canvas' button widget with Channel-ID (1st parameter).
	Reports the state (true/false or 0/1) of the Widget Canvas' toggle widget with Channel-ID (1st parameter).

	<p>Reports the state of the widget canvas' multibuttoned widget with Channel-ID (1st parameter), as an axis. This means: If the lower button is pressed, it reports -100. If the upper one is pressed, it reports 100. If neither is pressed, it reports 0.</p>
	<p>Instructs the widget canvas' speedometer, integer-display or any other display with Channel-ID (1st parameter) to show the value (2nd parameter).</p>
	<p>Reports the position (-100 to 100) of the widget canvas' joystick widget with Channel-ID (1st parameter). The axis is chosen by (2<sup>nd</sup> parameter): If it's 0, the block will report the horizontal axis. If it's 1, the block will report the vertical axis. Any other value will stop the block from reporting values.</p>
	<p>Reports the position (- 100 to 100) of the widget canvas' slider widget with Channel-ID (1st parameter).</p>
	<p>Reports the amplitude (- 100 to 100) of the widget canvas' joystick widget with Channel-ID (1st parameter).</p> <p>The further away the joystick is pulled from the center, the higher the number will be. In the upper half of the joystick, the numbers reported will be positive. In the lower half, they'll be negative.</p>
	<p>Reports the angle (normalized to -100 to 100 not in degrees, which would be -90 to 90!) of the widget canvas' joystick widget with Channel-ID (1<sup>st</sup> parameter).</p> <p>If the joystick points directly up or down (regardless of amplitude), the block outputs 0. If the joystick's handle is then motioned clockwise, the number will become negative and decrease until it reaches -100 at directly left or right. If rotated counterclockwise, the number is positive and increases towards 100.</p>