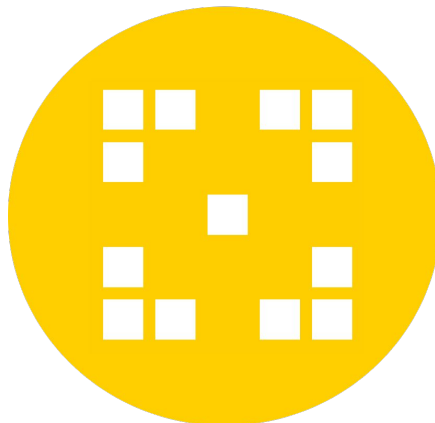


# *Unofficial SPIKE Prime Block Guide by PrimeLessons.org*

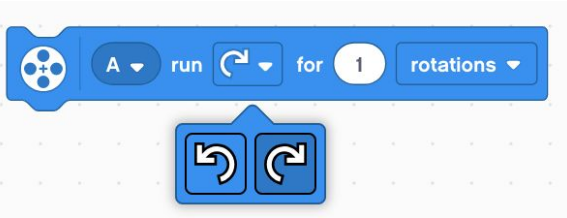


- Text descriptions based on SPIKE Prime Help Menu, but modified where necessary. Some blocks in the Help are not in the software. Other text in the Help does not match the actual block.
- To download additional programming blocks, click on the icon with blocks and a plus sign at the bottom left of the SPIKE Prime App and add the Extensions.

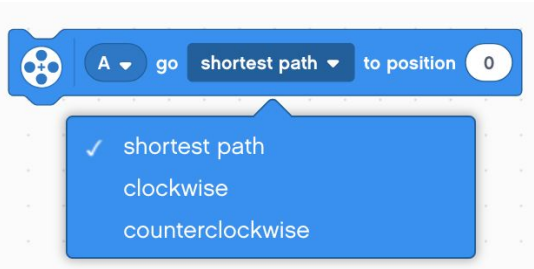




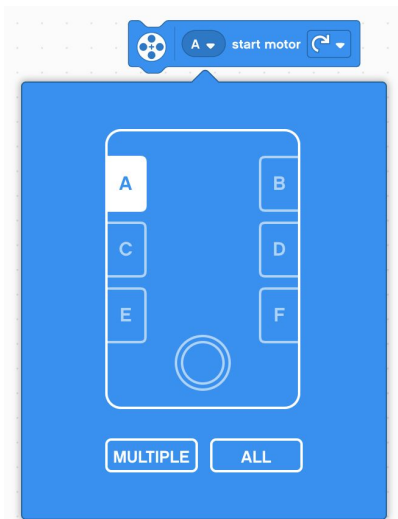
# MOTOR BLOCKS



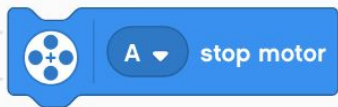
**Run Motor for Duration:** Tells the motor(s) to run in a clockwise or counterclockwise direction for a number of rotations, seconds or degrees. (Default speed: 75%, and Stall Detection enabled).



**Motor Go to Position:** Tells the motor(s) to travel the shortest path, clockwise or counterclockwise to the position selected (0-360). (Default speed: 75%, and Stall Detection enabled).



**Start Motor:** Starts the motor(s) turning in a clockwise or counterclockwise direction. (Default speed: 75%, and Stall Detection enabled).



**Stop Motor:** Stops the motor(s) selected. The motor will brake, and will not post the position



**Set Motor Speed:** Sets the speed of the motor(s) to the maximum percentage (-100 to 100). Negative value reverses direction.



**Motor Position:** Reports the current position of the selected motor (0-359).

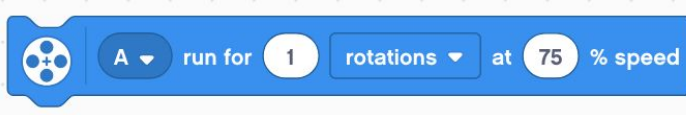


**Motor Speed:** Reports the actual current speed of the motor (-100-100).

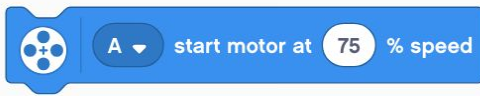


# MORE MOTOR BLOCKS

You will need to add these blocks using Extensions.



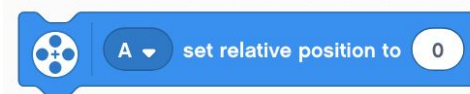
**Run Motor for Duration at Speed:** Runs motor(s) clockwise or counterclockwise for a number of rotations, seconds, or degrees at a specified speed (as a percentage or rpm). Stall detection is enabled.



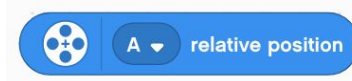
**Start Motor at Speed:** Runs motor(s) clockwise or counterclockwise forever at the specified speed (percentage or rpm). Stall detection is enabled.



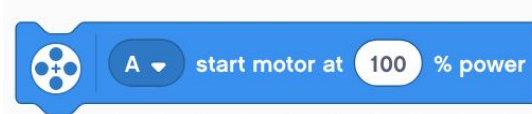
**Go to Relative Position at Speed:** Runs motor(s) to a relative position at the specified speed (%). Stall detection is enabled.



**Set Relative Motor Position to 0:** Sets the relative position of selected motor(s) to a specified value. "0" resets the relative position.



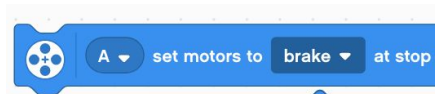
**Relative Motor Position:** Reruns the number of degrees the motor has turned since the program started or was reset to 0.



**Start Motor with Power:** Runs motor(s) at specified percentage of power forever. Stall detection enabled.

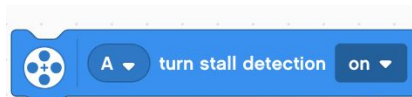


**Motor Power:** Returns the power level being used on a specified motor (in %)



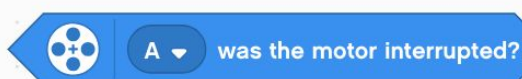
✓ brake  
hold position  
coast

**Stop and Coast Motors:** Defines how the motor with stop. Brake (default method applied friction to the motor). Hold (uses power to brake and moves motor back to the position it was stopped at). Coast (the power to the motor is cut when stopping)



✓ on  
off

**Turn Stall Detection On/Off.** Use this to disable Stall detection on motor and movement blocks. If stall detection is off, the motor will keep trying to complete even when physically prevented. If on, the code will move on to the next block.

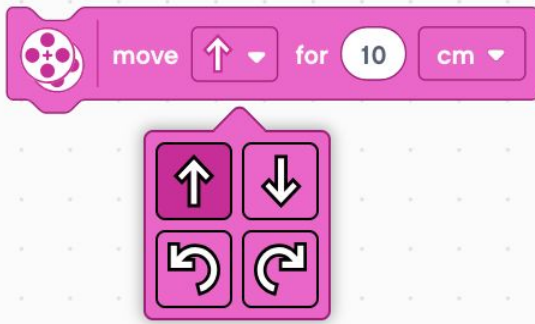


**Was Movement Interrupted?** If a motor block with a specified duration did not complete, it was interrupted.

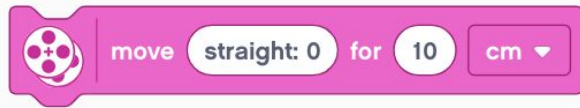


# MOVEMENT BLOCKS

Movement Block motors are synchronized. They must be two of the same type.



**Move for Duration:** Move forward, backwards, left or right for centimeters, inches, rotations, degrees, or seconds. Distance moved in cm/in depends on what the “Set 1 Rotation to Distance Moved” block is defined as.



**Move with Steering for Duration:** Move forward, backwards, left or right for centimeters, inches, rotations, degrees, or seconds at a given steering. Left (-100), Right (100), Straight (0)



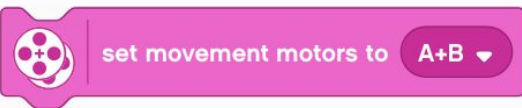
**Start Moving with Steering:** Starts the motors moving at a steering. Left (-100), Right (100), Straight (0)



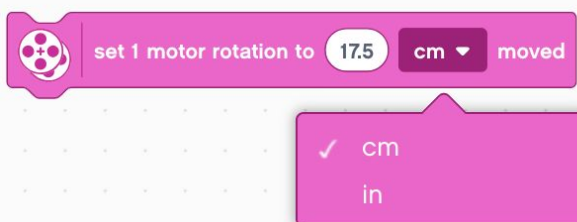
**Stop Moving:** Stops the motor(s) moving



**Set Movement Speed:** Sets the motors to move at a percentage of their maximum speed by default(-100 to 100).



**Set Movement Motors:** Defines which two motors are used for movement for your driving base



**Set 1 Rotation to Distance Moved:** Defines how many cm or inches one motor rotation equals.

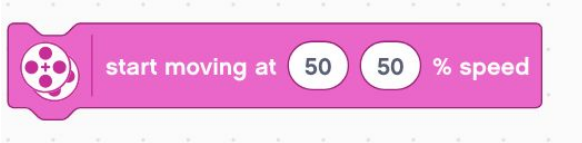


# MORE MOVEMENT BLOCKS

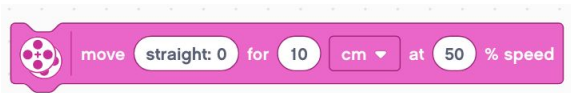
You will need to add these blocks using Extensions.



**Move for Duration at Speed:** Moves the driving base for specified centimeters, inches, secs, degrees or rotations at a specified speed for left and right motors.



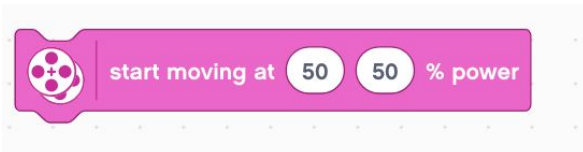
**Start Moving at Speed:** Moves the driving base forever at a specified speed for left and right motors.



**Move for Duration with Steering at Speed:** Moves the driving base for a specified duration (cm, in, sec, degrees or rotations) at a specified speed while steering. Left (-100), Right (100), Straight (0)



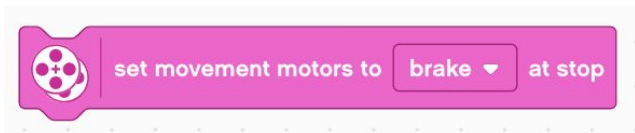
**Start moving with Steering at Speed:** Moves the driving base at specified speed while steering left (-100), right (100) or straight (0) forever.



**Start Moving with Power:** Move driving base forever at the specified power for left and right motors. Speed will not be regulated.



**Start Moving with Steering and Power:** Moves the driving base at specified power (unregulated) while steering left (-100), right (100) or straight (0) forever.



## Set Movement Motors to Brake at stop

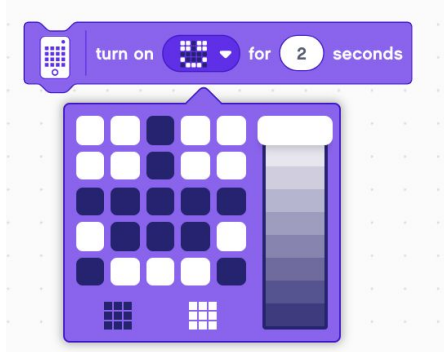
Determines how the motors will stop when using a Movement Block for a specified duration. Brake (Power to brake), Hold (Power to break and motor moves back to the position when it stopped), Coast (Power to the motor is cut)



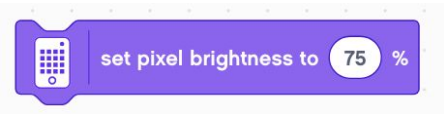
**Did movement get interrupted?** Returns true if the movement of the last motor block with a specified duration did not complete because a programming block interrupted it.



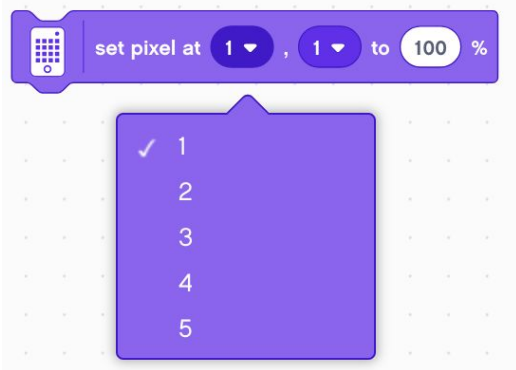
# LIGHT BLOCKS



**Turn on 5X5 Light matrix for Seconds:** Create a pattern that lights up for a specific amount of time. Move the lever to change the intensity of the light.



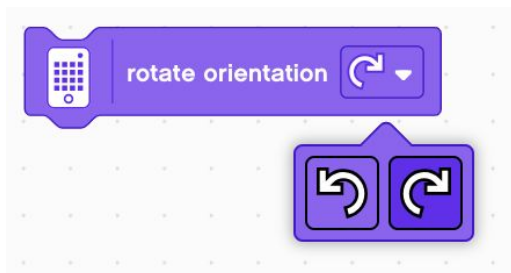
**Set Pixel Brightness:** Set the brightness of the 5X5 Light Matrix for the next block using the 5X5 Light Matrix. Default: 100%



**Set Pixel:** Set the brightness of individual pixels on the 5X5 Light Matrix.



**Set Orientation to (Upright):** Set the orientation of what is being shown on the Light Matrix. Choose between upright, upside down, left or right. Default: Upright.

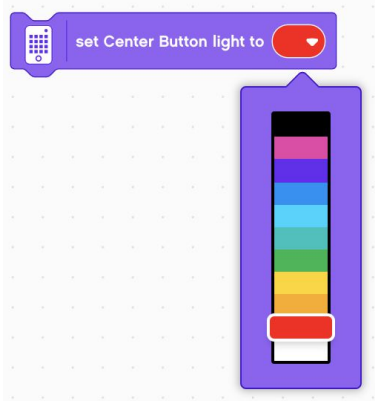


**Rotate Orientation (Clockwise):** Rotate the orientation of what is being shown on the 5X5 Light Matrix to clockwise or counterclockwise.

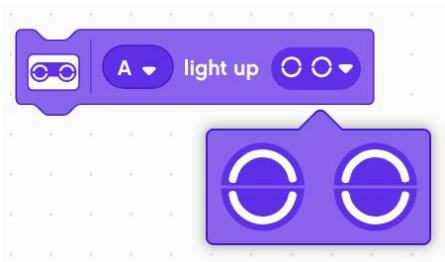




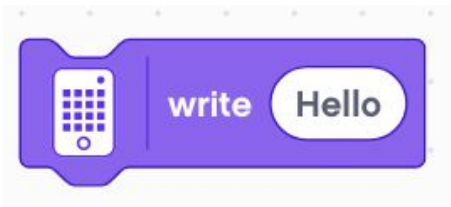
# LIGHT BLOCKS



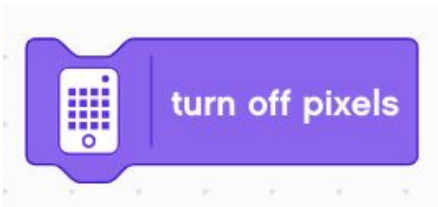
**Set Center Button Light:** Set the color of the Center Button light



**Light up Distance Sensor:** Turn on the lights on the four segments of the Distance Sensor



**Write on 5X5 Matrix:** Display a text string on the 5X5 Light Matrix that scrolls by



**Turn off Pixels:** Turn off the all the lights on the 5X5 Light Matrix.



**Turn on 5X5 Light Matrix:** Create a pattern to light up on the Light Matrix. Pattern stays light until the Light Matrix is told to do something else or the program is stopped.

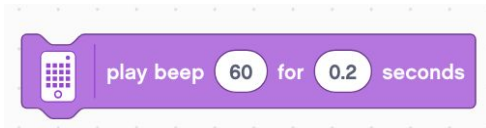


# SOUND BLOCKS

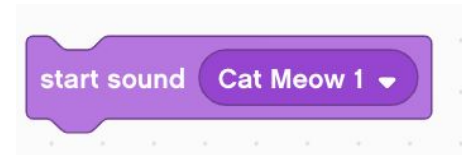
Most sounds (other than beeps) play on your device and not the Hub.



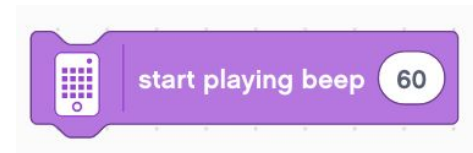
**Play Sound until Done:** Plays a selected sound on your device and pauses the programming stack until the sound is finished. You can add sounds, record sounds and edit sounds.



**Play Beep for Seconds:** Plays a beep tone on the Hub for the specified number of seconds.



**Start Sound:** Start playing a selected sound on your device and immediately plays the next block in the programming stack. You can add sounds, record sounds and edit sounds.



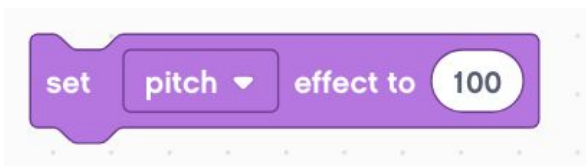
**Start Playing Beep:** Plays a beep tone on the Hub until something in the program stops it.



**Stop All Sounds:** Stops all sounds currently being played.



**Change Pitch Effect By:** Changes the pitch or pan left/right effect of the sounds played on the device. Pan effect: which speaker is emitting the sound. Left Speaker (-100), Normal (0), and Right Speaker (100)



**Set Pitch Effect By:** Changes the pitch or pan left right/left effect of sounds that are being played on the device.





# SOUND BLOCKS



clear sound effects

**Clear Sound Effects:** Sets both the pitch and pan left/right sound effect back to normal



change volume by -10

**Change Volume:** Changes the volume of the sound currently being played by a specified increment from the volume at which it's currently playing. Default (100%)



set volume to 100 %

**Set Volume:** Sets the volume of the sound. Default (100%)



volume

**Volume:** Reports the current volume



# EVENT BLOCKS

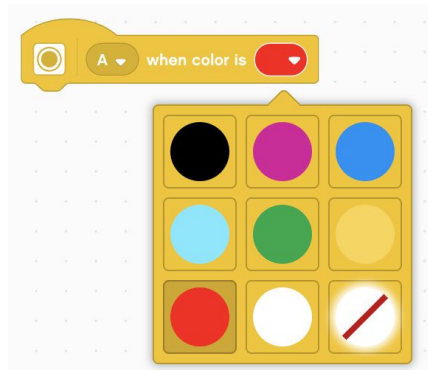
**When Program Starts:** Plays all the blocks attached to it in sequence

**When Color Is:** Plays all the blocks attached to it when the Color Sensor detects a specified color

**When Pressure Is:** Plays all the blocks attached to it when the Force Sensor is pressed, hard-pressed, released, or when a change in pressure is detected.

**When Closer Than:** Plays all the blocks attached to it when the Distance Sensor detects that an object is closer than or further than the specified distance.

**When Hub Orientation is Up:** Plays all the blocks attached to it when the Hub is placed in the specified orientation (front, back, top, bottom, left side, or right side)





# EVENT BLOCKS

**When Hub Shaken:** Plays all of the blocks attached to it when the hub is shaken, tapped, or falling.

**When Hub Button Pressed:** Plays all the blocks attached to it when the Left or Right Buttons are pressed or released.

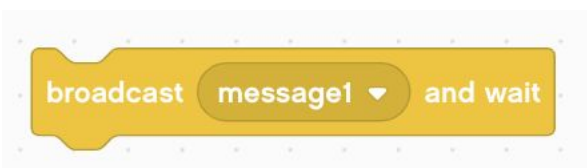
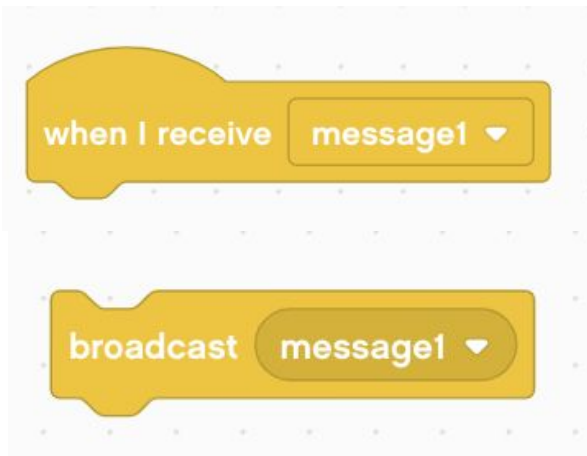
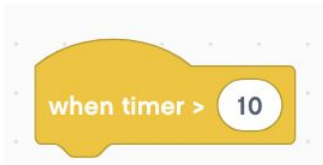
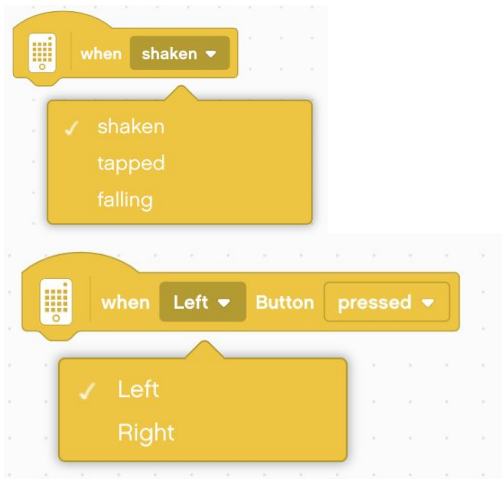
**When Timer:** Plays all the blocks attached to it when the time exceeds the specified value.

**When:** Plays all the blocks attached to it when a certain condition is true

**When I receive Message:** Plays all the blocks attached to it when a specified message is broadcasted by the Broadcast Message Block or the Broadcast Message and Wait Block.

**Broadcast Message:** Broadcasts a specified message. All When I Receive Message Hat Blocks for that specified message will play. After the message has been sent, the next block in the programming stack with play.

**Broadcast Message and Wait:** Broadcasts a specified message. All When I Receive Message Hat Blocks for that specified message will play. After the message has been sent, the block waits until all the programming stacks with the specified message finish before moving to the next block.





# CONTROL BLOCKS



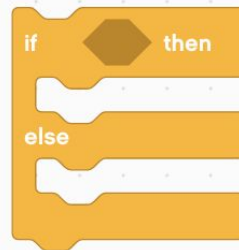
**Wait for Seconds:** Pauses the stack for the specified number of seconds (whole numbers and decimals)



**Repeat Loop:** All blocks in the loop will repeat for the specified number of times



**If Then:** The block checks whether a condition is true. If true, everything inside will play.



**If Then Else:** The Block checks whether a specific condition is true. If true then the the blocks in the first space will play and then the code continues down the stack. If false, the blocks in the second space play.



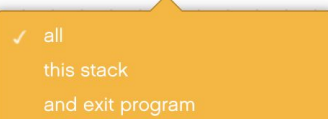
**Wait Until:** Pauses the stack until the condition is true



**Repeat Until Loop:** All blocks inside the loop will repeat until the condition is true



**Stop other Stacks:** The block stops all other programming stacks in the project except its own



**Stop:** Block stops all programming stacks currently running, it's own programming stack, or exits the program.



# SENSOR BLOCKS



**Is Color?:** Returns “true” when the Color Sensor detects the specified color.



**Color:** Reports the color of the Color Sensor as a number code.



**Is reflected light:** Returns “true” when the light reflected back to the Color Sensor is greater than, equal to, or less than the specified percentage.



**Reflected Light:** Reports the value of the light that is being reflected back to the Color Sensor



**Is Pressed?** Returns “true” when the Force Sensor is pressed (> 0 newton), hard pressed (> 5 newton), or released (= 0 newton)



**Pressure:** Reports the current pressure being applied to the Force Sensor in newtons (2-10 newtons) or as a percentage.



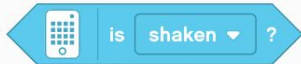
**Is distance?** Returns “true” when the Distance Sensor detects something is closer than, exactly or greater than a specified distance.



**Distance:** Reports current distance the Distance Sensor is detecting (cm, in, percentage).



**Is Hub orientation?** Returns “true” if Hub is placed at angle specified (front, back, top, bottom, left side, right side)



**Is Shaking?** Returns “true” when the Hub is shaken, tapped, or falling.



**Hub Pits Roll Yaw Angle:** Reports the Hub's pitch, roll or yaw angle.



**Set Hub Yaw Angle to 0:** Sets the yaw = 0



**Is Hub Button pressed?** Returns “true” if Left or Right button is pressed or released



**Time:** Reports time (sec) since the program started.

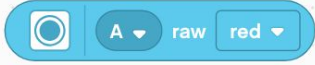


**Reset Timer:** Resets the timer



# MORE SENSOR BLOCKS

You will need to add these blocks using Extensions.



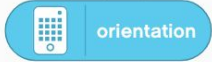
**Raw Color:** Returns the raw red, green, or blue color reading from the Color Sensor (0-255)



**Hub Acceleration:** Returns the Hub's acceleration on the X, Y or Z axis



**Hub Angular Velocity:** Returns the Hub's angular velocity (Gyro Rate) on the X, Y, or Z axis



**Orientation:** Returns the current orientation of the hub (front, back, top, bottom, left side, or right side)



**Gesture:** Returns the current gesture (shaken, tapped, or falling)

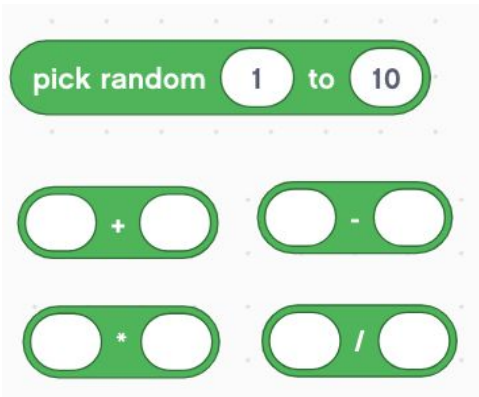


**Set Hub Sensor Orientation:** Set the orientation of the 6-axis Gyro Sensor to front, back, top, down, left side, or right side.



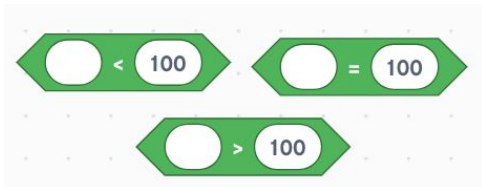


# OPERATOR BLOCKS

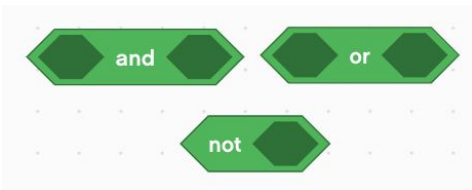


**Pick Random Number:** Picks a random number within the specified range (including endpoints). Whole numbers or decimals can be selected if the endpoints are decimals. If endpoints are whole numbers, only whole numbers can be selected

**Plus/Minus/Multiply/Divide:** Add, subtract, multiply or divide two values and return the results



**Greater Than/Less Than/Equal To:** Returns true if value is greater than, less than or equal to.



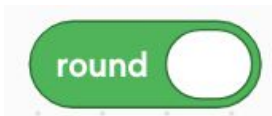
**And/Or/Not:** Joins two Boolean Blocks with “and”, “or” condition. Not inverts the boolean value of the condition inside.



**Is Between:** Checks whether the specified value falls between the next two specified values (including endpoints)



**Mod:** Returns the remainder when the first value is divided by the second. E.g. 10 mod 3 returns 1.



**Round:** Rounds the given number to the nearest integer. (.5 and higher round up)



# OPERATOR BLOCKS



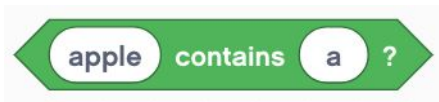
**Join Strings:** Join two text values and returns the results. E.g. “apple” “banana” would return “applebanana”



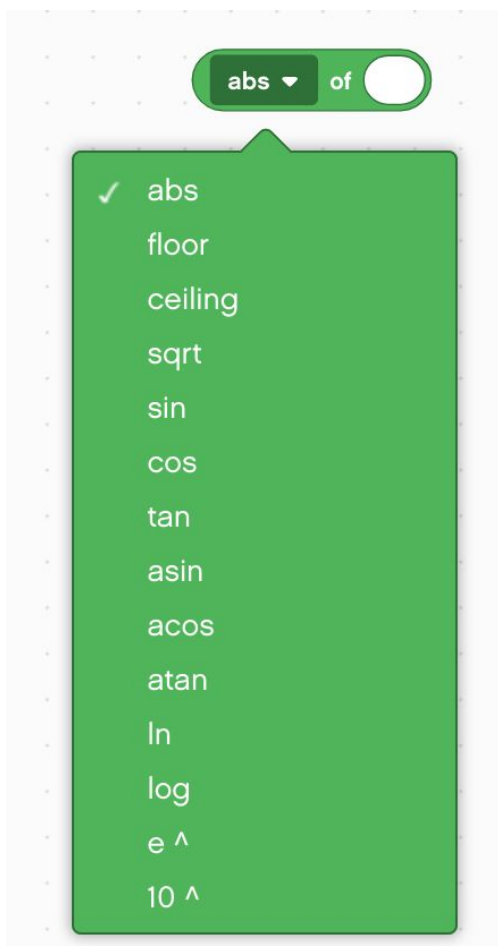
**Letter of String:** Returns the character that occupied the specified position. E.g. letter 1 of apple is “a”



**Length of String:** Returns the number of characters in the string. E.g. If you enter “apple”, the block returns “5”



**String Contains:** Returns “true” if the specified character is contained in the string.



**Math Functions:** Performs the specified math function on a given number and reports the results.