

## レゴ®Powered UP プログラミングブロックガイド

シンプルでわかりやすいプログラミング言語が特徴のレゴ®Powered UP は、幅広い年齢層のレゴファンにとって親しみのあるブロックのプログラミング言語を使用しているので、レゴブロックをカチッとはめる感覚で、楽しみながらプログラミングを学ぶことができます。







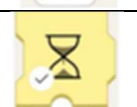

プログラミングブロックには様々な種類と色がありますが、使用するブロックごとに色分けされています。






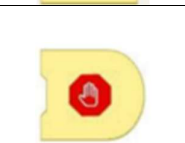
よく使われているプログラミングブロックの例を紹介します。

遊びながら学んでいくのはレゴ Powered UP ならではの。プログラミングブロックの動作がわからないときや、ここで紹介されていないブロックは、ぜひ自分で試して動作を確認してみてください。

### 黄色のフローブロック



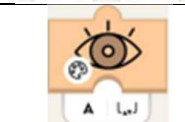
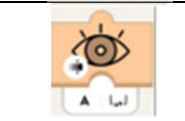

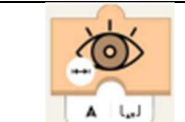
黄色のブロックはプログラミングの流れを制御します。プログラムの開始、停止、一時停止、命令を繰り返すときに使用します。

	プログラムを開始させます。
	真の状態になった時にプログラミングを開始させます。
	偽から真の状態になった時にプログラミングを開始させます。
	フラグ番号が誘因されたときにプログラムを開始させます。
	フラグ番号を誘因します。
	指定した秒数だけ、プログラムを待機させます。
	真の条件になるまで待機させます。
	指定した回数分、動作を繰り返します。

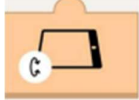
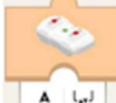
	真の条件の間、動作を繰り返します。
	プログラムを無限に繰り返します！
	条件が真の場合に上位の配列動作を実行します。上位の配列が真でない場合には下の配列動作を実行します。
	両方の配列を同時に作動させます。両方の配列が終了するまで次のプログラミングブロックへは移行しません。
	他のプログラムをすべて停止させ、次に続くコードの流れに沿って動作をします。
	キャンバスで実行されているすべてのプログラムを停止させます。 もし最上位のプログラムに使われている場合はプログラムを終了します。モデルブロックに使用され、モデルブロックが終了した場合、モデルブロックを含むプログラムは引き続き実行されます。

## オレンジ色のセンサブロック

オレンジ色のブロックは、色と距離センサー、ムーブハブのチルトセンサー、リモコン、WeDo 2.0のセンサーと連動します。モデルのセンサーが何かを感知した際にプログラミングが作動します。

	カラーセンサーで指定された色を検知したときに動作を誘因します。
	カラーセンサーが指定された色を検知するまで待機します。
	センサーで測定された色を取得します。
	センサーで明るさを測定し、取得します。
	設定距離より近い距離を測定したら、動作を開始させます。
	センサーで測定された距離を取得します。


	設定距離より近い距離を測定するまで待機します。
	センサーで動力を測定し、接続されたブロックの動作に作用させます。
	内臓のジャイロスコープを使って、測定した傾きに基づいた動作を開始させます。
	設定した傾きとハブの傾きが等しくなるまで待機します。
	ハブの傾きを取得します。
	ハブの X 軸方向の傾きを取得します。
	ハブの Y 軸方向の傾きを取得します。
	指定した軸の傾きをチルトセンサーで取得します。
	指定した軸の傾きをハブ内臓の加速度センサー/チルトセンサー取得します。
	チルトセンサーの方向を設定します。
	(WeDo センサー専用)距離を測定します。
	(WeDo センサー専用) Y 軸上で WeDo センサーチルトの傾きを取得します(-90/90)。
	(WeDo センサー専用) X 軸上で WeDo センサーチルトの傾きを取得します(-90/90)。
	チルトセンサーによって測定された端末機/タブレットの X 軸(左右方向)を使用します。0 は中心軸です。

	チルトセンサーによって測定された端末機/タブレットの Y 軸(上下方向)を使用します。0 は端末スクリーンを正面に向かって 30 度ほど傾けた位置です。
	接続されたリモコンの A 側または B 側で押されたボタンを検出します。

## 緑色のモーターまたは移動ブロック











緑色のブロックがモデルを動かします！タコモーター（速度と位置の制御用）とベーシックモーター（全速力駆動用）を区別して使用しましょう。

	出力を設定します。正の数は時計回りに、負の数は反時計回りに回転します。
	モーターへの出力を停止し、モーターを徐々に停止させます。
	モーターを強制停止させます。
	指定したポートに繋がっているモーターの速度値を検出します。
	指定したポートに繋がったタコモーターを指定された速度で動かします。
	指定したポートに繋がったタコモーターを指定された速度で指定した時間（秒）動かします。
	タコモーターの位置を検出します。位置は相対位置で、0 はモーターに電源が入った位置です。
	モーターを指定角度に位置設定します。例：モーターの角度 $45^{\circ}$ + $180^{\circ}$ の角度設定 = $225^{\circ}$ （モーターの最終角度）
	モーターを指定角度まで位置設定します。例：モーターの角度 $45^{\circ}$ → $180^{\circ}$ （指定角度）
	指定されたポートのタコモーターの現在位置を指定された位置に設定します。
	指定されたポートのタコモーターを停止した状態で保ちます。

	指定されたポートでタコモーターへの出力上限を設定します。
	2つのタコモーターへの出力を個別に設定します。
	2つのタコモーターの速度を個別に設定します。
	2つのタコモーターの速度と回転時間(秒)を個別に設定します。
	2つのタコモーターの位置を個別に設定します。
	指定されたポート (AB または CD) で一対のタコモーターの速度を設定し、ステアリング方向はタンクステアリング形式の動作で定義します。0 は両方のモーターへ等しく出力し、50 は片方のモーターにのみ出力し、100 はモーターに逆方向に出力します。ステアリングの正負値で方向を定義します。
	指定されたポート (AB または CD) で一対のタコモーターの速度を設定し、ステアリングの向きはタンクステアリングの動作で定義します。モーターは指定された時間 (秒) だけ作動します。0 は双方に等しい動力を送り、50 は片方のモーターにのみに動力を送ります。100 はモーターに反対方向の動力を送ります。正負値はステアリングの方向を定義します。
	指定したポートで一対のタコモーターの速度を設定し、ステアリングの向きはタンクステアリングの動作で定義します。モーターは指定の角度に達するまで回転します。0 は双方に等しい動力を送り、50 は片方のモーターにのみに動力を送ります。100 はモーターに反対方向の動力を送ります。正負値はステアリングの方向を定義します。
	指定されたポートのタコモーターの角度を設定します。
	指定されたポートに接続されたモーターへの動力を設定し、ループ設定なしでプログラムが終了されるまで実行します。正の値は時計回りに、負の値は反時計回りに回転します。
	指定されたポートのタコモーターの加速時間 (秒) を設定します。
	指定されたポートのタコモーターの減速時間 (秒) を設定します。

## 紫色のサウンドとライトのブロック





紫色のブロックは、端末（スマートデバイス）のスピーカーからサウンドを再生したり、ムーブハブ、色や距離センサーのライトの色を制御したりすることができます。



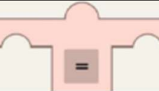

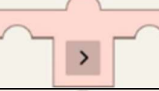

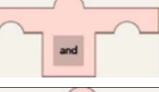
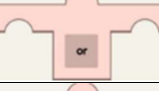
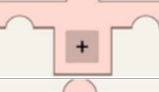
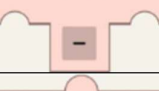
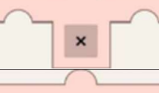
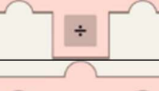
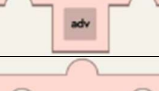


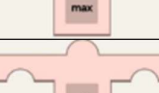


	ポートに接続されているカラーセンサーと距離センサーのモードを設定します。8 は接続時の初期値、7 は赤外線モードです。
	センサーLED の色を設定します。
	ハブ LED の指定し、色を設定します。
	ポートに接続されているライトの明るさを設定します。
	<p>接続されたポート(左のパラメーター)から出力される明るさを設定しますが、センサーにある 4 つの LED の半円は個別に設定します。</p> <p>左から右の順で、以下に影響します。</p> <ul style="list-style-type: none"> <li>・ 右「目」上部の円弧</li> <li>・ 左「目」上部の円弧</li> <li>・ 右「目」下部の円弧</li> <li>・ 左「目」下部の円弧</li> </ul>
	指定のポートに接続された色と距離センサーで、パワーファンクションの IR レシーバーに IR コマンドを送信します。最初の入力スロットは、センサーが接続されているポートです。2 番目は IR ポート 1～4 にマッチする 0～3 の IR チャンネルです。3 番目はいくつかを定義しますが、基本的な使い方は、4 で赤、5 で青のポートをコントロールします。4 はスピードを設定し、0 はフロート（スピードなし）、1 から 7 は前進増速、8 は停止、9 から F は後進減速となります。
	指定された RGB 強度（赤、緑、青の値を%で表示）に応じて、ハブ LED が色を表示するように設定します。
	ハウスサウンドライブラリから選択したサウンドを再生します。
	ホイッスルサウンドライブラリから選択したサウンドを再生します。
	列車サウンドから選択したサウンドを再生します。

	車サウンドライブラリから選択したサウンドを再生します。
	飛行機サウンドライブラリから選択したサウンドを再生します。
	犬サウンドライブラリから選択したサウンドを再生します。
	猫サウンドライブラリから選択したサウンドを再生します。
	おならサウンドライブラリから選択したサウンドを再生します。
	笛サウンドから選択したサウンドを再生します。
	雑踏サウンドライブラリから選択したサウンドを再生します。
	ロボットサウンドライブラリから選択したサウンドを再生します。
	ターゲットサウンドライブラリから選択したサウンドを再生します。

## ライトグレーの算数ブロック

ライトグレーのブロックでは、変数を使った複雑なプログラミングができます。

	左のスロットに特定のハブを選択し、右に入力されたポートから出力します。
	ローカル変数を取得します。
	ローカル変数を保存された数値に設定します。
	グローバル変数を取得します。

	グローバル変数を保存された数値に設定します。
	数値と数値の間のランダムな値を出します。(設定された二つの数字を範囲に含む)
	数値同士が等しいかを返します。
	数値を比較して左の数値が右の数値より小さいかを返します。
	数値を比較して左の数値が右の数値より大きいかを返します。
	数値同士が不一致かを返します。
	双方の数値が真の場合に真を返します。それ以外の場合には偽を返します。 例：入力値（左と右）の両方が真なら真を、そうでない場合に偽を返します。
	双方のいずれかの数値が真の場合に真を返します。それ以外の場合には偽を返します。
	数値同士を足した値を返します。
	数値同士を引いた値を返します。
	数値同士をかけた値を返します。
	左の数値を右の数値で割った値を返します。
	数種類の演算を使用して計算された値を返します。例：算術演算子（左側）と数値（右側）から指定した演算を使用して値を返します。
	位置の角度(左右の数値)を表す $-\pi$ と $\pi$ の間の数値を返します。
	左に入れた数を右に入れた数の回数だけ掛けた結果を返します。
	最高値を返します。
	最低値を返します。
	左の数値を右の数値で割り、その余りを導き出します。

## 黄緑色のウィジェットブロック

黄緑色のブロックはウィジェットを使ってモデルを操作します。

	チャンネル ID (1 番目のパラメータ)を持つウィジェットキャンバス；ボタンの状態 (真・偽または 0/1)を取得します。
	チャンネル ID (1 番目のパラメータ)を持つウィジェットキャンバスのトグルウィジェットの状態(真/偽または 0/1)を報告します。
	チャンネル ID (1 番目のパラメータ)を軸として、ウィジェットキャンバスのマルチボタンウィジェットの状態を報告します。つまり下のボタンが押された場合、-100 を報告します。上のボタンが押された場合、100 を報告します。どちらも押されていない場合は、0 を報告します。
	ウィジェットキャンバスの速度計、整数ディスプレイ、またはチャンネル ID (1 番目のパラメータ)を持つその他のディスプレイに、値(2 番目のパラメータ)を表示するように指示します。
	ウィジェットキャンバスのジョイスティックウィジェットの位置(-100~100)を チャンネル ID(1 番目のパラメータ)で報告します。軸は(2 番目のパラメータ)で選択されます：0 であれば、ブロックは横軸を報告します。1 の場合、ブロックは縦軸を報告します。それ以外の値を指定すると、ブロックは値を報告しません。
	チャンネル ID (1 番目のパラメータ)を持つウィジェットキャンバスのスライダーウィジェットの位置(-100 から 100)を報告します。
	ウィジェットキャンバスのジョイスティックウィジェットのチャンネル ID(1 番目のパラメータ)の振幅(-100 から 100)を報告します。 ジョイスティックが中心から離れるほど、数値は高くなります。ジョイスティックの上半分では、報告される数字は正の数になり、下半分では負の数になります。
	チャンネル ID (1 番目のパラメータ)を持つジョイスティックウィジェットの角度を報告します(度単位ではなく、-100 から 100 に正規化され、-90 から 90 になります!)。ジョイスティックが(振幅に関係なく)真上または真下を向いた場合、 ブロックは 0 を出力します。その後ジョイスティックのハンドルを時計回りに動かすと、 数値はマイナスになり、左または右で-100 になるまで減少します。反時計回りに回転させると、数値は正の数になり、100 に向かって増加します。