










Scratch ブロックの解説















ブロックの詳しい解説です。オリジナルプログラムを作るときの参考にしてね。

















動き	
 10 歩動かす	スプライトを前後に動かす
 15 ↻ 度回す	スプライトを時計回りに回す
 15 ↺ 度回す	スプライトを反時計回りに回す
 90 ▾ 度に向ける	スプライトを指定された方向に向ける 0 の場合は上、90 の場合は右、180 の場合は下、-90 の場合は左
 ▾ へ向ける	スプライトをマウスポインタや他のスプライトの方向に向ける
 ×座標を 0、y座標を 0 にする	スプライトを、ステージ上の指定された X、Y 座標に移動する
 ▾ へ行く	スプライトを、マウスポインタや他のスプライトの場所に移動する
 1 秒で×座標を 0 に、y座標を 0 に変える	スプライトを、指定された場所へ指定された時間でなめらかに移動する
 ×座標を 10 ずつ変える	スプライトの X 座標を指定された数ずつ変更する
 ×座標を 0 にする	スプライトの X 座標を指定された値に設定する
 y座標を 10 ずつ変える	スプライトの Y 座標を指定された数ずつ変更する
 y座標を 0 にする	スプライトの Y 座標を指定された値に設定する
 もし端に着いたら、跳ね返る	ステージの端についたとき、スプライトを反対方向に向ける
 ×座標	スプライトの X 座標を示す (-240 から 240 までの間)
 y座標	スプライトの Y 座標を示す (-180 から 180 までの間)





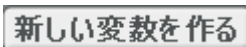




向き	スプライトの向きを示す 0 の場合は上、90 の場合は右、180 の場合は下、-90 の場合は左
見た目	
コスチュームを コスチューム2 にする	スプライトの見た目を、異なるコスチュームに切り替える
次のコスチュームにする	スプライトのコスチュームを、コスチュームリストの次のものに変える(次がリストの最後だった場合、最初に戻る)
こんにちは! と 2 秒言う	スプライトのふきだしを、指定された時間だけ表示する
こんにちは! と言う	スプライトのふきだしを表示する(空のテキストを持つブロックを実行すると、ふきだしは消える)
うーん... と 2 秒考える	スプライトの考えごとのふきだしを、指定された時間だけ表示する
うーん... と考える	スプライトの考えごとのふきだしを表示する
色 の効果を 25 ずつ変える	スプライトの表示効果を指定された量だけ変更する(プルダウンメニューを使って効果を選択する)
色 の効果を 0 にする	視覚効果を指定された数に設定する ほとんどの視覚効果の範囲は 0 から 100 である
画像効果をなくす	すべての視覚効果をスプライトから消去する
大きさを 10 ずつ変える	スプライトの大きさを指定された値ずつ変更する
大きさを 100 % にする	スプライトを、もとの大きさの指定された割合に設定する
大きさ	スプライトの大きさを、もとの大きさとの割合で表示する
表示する	スプライトをステージ上に出現させる

隠す	スプライトをステージ上から消す。隠れているとき、他のスプライトは「触れた」ブロックで、このスプライトを見つけられない
前に出す	スプライトを、他のすべてのスプライトの手前に移動する
1番目に移動	スプライトを、指定されたレイヤーの数だけ奥に移動する。これにより、スプライトは他のスプライトの背後に隠れる
音	
ニャーの音を鳴らす	プルダウンメニューで選択された音を再生し、音を鳴らしたまま、すぐ次のブロックに処理を進める
終わるまでニャーの音を鳴らす	音を再生し、音が終わるまで待ってから次のブロックへ進む
48のドラムを0.2拍鳴らす	プルダウンメニューで選択したドラムの音を、指定された拍子で鳴らす
60の音符を0.5拍鳴らす	音符（数値が大きいほど音程が高い）を指定された拍子で鳴らす
楽器を1にする	スプライトが音符を鳴らすブロックを実行するときを使う楽器を設定する（すべてのスプライトは、独立した楽器を持つ）
すべての音を止める	すべてのスプライトのすべてのスクリプトの実行を止める
ペン	
消す	ステージから、すべてのペンの跡とスタンプを消す
ペンを下ろす	スプライトのペンを下ろし、移動したとき線がひけるようにする
ペンを上げる	スプライトのペンを上げ、移動したとき線が引けないようにする
ペンの色を■にする	色を選択するツールで、ペンの色を設定する
ペンの色を10ずつ変える	ペンの色を指定された量ずつ変更する

	ペンの色を指定された値に設定する(ペンの色 0 は虹色パレットの赤色の端、100 は青色の端)
	ペンの濃さを指定された量ずつ変更する
	ペンの濃さを指定された値に設定する(ペンの濃さ 0 は一番濃く、100 は一番薄い)
	ペンの太さを指定された値ずつ変更する
	ペンの太さを指定された値に設定する
	スプライトの画像をステージ上にスタンプする
制御	
	緑の旗がクリックされたときに、下に続くスクリプトを実行する
	指定されたキーが押されたときに、下に続くスクリプトを実行する
	スプライトがクリックされたときに、下に続くスクリプトを実行する
	指定された秒数だけ待ち、その後で次のブロックの実行を続ける
	中に入っているブロックを永遠に繰り返して実行する
	中に入っているブロックを指定された回数だけ実行する
	すべてのスプライトに指定された処理を起動するメッセージを送り、すべてのスプライトの実行が終わるまで待ってから、次のブロックの実行を続ける
	すべてのスプライトに指定された処理を起動するメッセージを送り、すべてのスプライトの実行が終わるのを待たずに次のブロックの実行を続ける

	指定されたメッセージが送られたときに、下に続くスクリプトを実行する
	条件が心かどうかを確認し、真である限り、中に入っているブロックを実行する
	条件が真のとき、中に入っているブロックを実行する
	条件が真のときは「もし なら」の方に入っているブロックを実行し、偽だったときは「でなければ」の方に入っているブロックを実行する
	条件が真になるまで待ち、真になったときに下に続くブロックを実行する
	条件が偽かどうかを確認し、偽だった場合には中に入っているブロックを実行して、また条件を確認する。条件が真になったとき、このブロックの下に続くブロックを実行する
	スクリプトの実行を止める
	すべてのスプライトのすべてのスクリプトの実行を止める
調べる	
	マウスポインタのX座標を示す
	マウスポインタのY座標を示す
	マウスポインタが押されたとき、真になる
	指定されたキーが押されたとき、真になる
	スプライトが、指定されたスプライトやステージの端、またはマウスポインタに触れたときに真になる(プルダウンメニューから選ぶ)
	スプライトが、指定された色に触れたときに真になる(色の部分をクリックすると出てくるスポイトを使って色を選ぶ)

	1番目の色（スプライトに含まれている色）が、2番目の色（背景や別のスプライト）に触れたときに真になる。 色の部分をクリックすると出てくるスポイトを使って色を選ぶ
	指定されたスプライトかマウスポインタからの距離を示す
	タイマーを0に設定する
	タイマーの値を秒で示す（タイマーはいつも動いている）
	コンピュータのマイクが捕らえた音量（1から100までの間）を示す
	コンピュータのマイクが30以上の音量（スケールは1から100まで）をとらえたときに真になる
	指定されたセンサーの値を示す
	指定されたセンサーが押されたときに真になる
演算	
	2つの数を足す
	1番目の数から2番目の数を引く
	2つ目の数をかける
	1つ目の数を2つめの数で割る
	指定された範囲の中から無造作に整数を取り出す
	1つ目の数が、2つ目の数より小さいとき真になる
	2つの数が同じとき真になる
	1つ目の数が、2つ目の数より大きいとき真になる
	2つの条件が両方とも真のとき真になる
	2つの条件のどちらかが真のとき真になる

	条件が偽のときに真になる。反対に、条件が真のとき偽になる。
	指定された数の絶対値を示す
	1 番目の数を 2 番目の数で割った余りを示す
	指定された数に最も近い整数を示す
変数	
	新しい名前で作数を作る。変数を作ったとき、3 つのブロックが自動的に作られる。(以下を参照)。その変数をすべてのスプライトで使えるか(グローバル)、1 つのスプライトだけで使えるか(ローカル)を選ぶ。
	変数に関連づけられた 3 つのブロックをすべて消す
	変数を指定された量ずつ変更する
	変数を指定された値に設定する
	変数の値を示す

応用編

変数をつくる

- ① コマンドグループパレットの「変数」をクリックします。
- ② ブロックパレット一番上の「新しい変数を作る」をクリックします。
- ③ 変数名につけたい変数名を入力し（この場合は Timer）、「OK」をクリックします。
すると図 3 のように変数が作成されます。

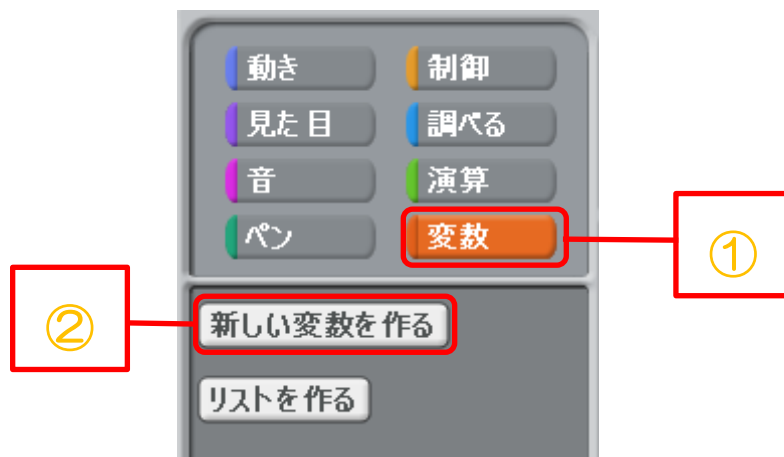


図 1 変数をつくる①

<次のページへつづきます>

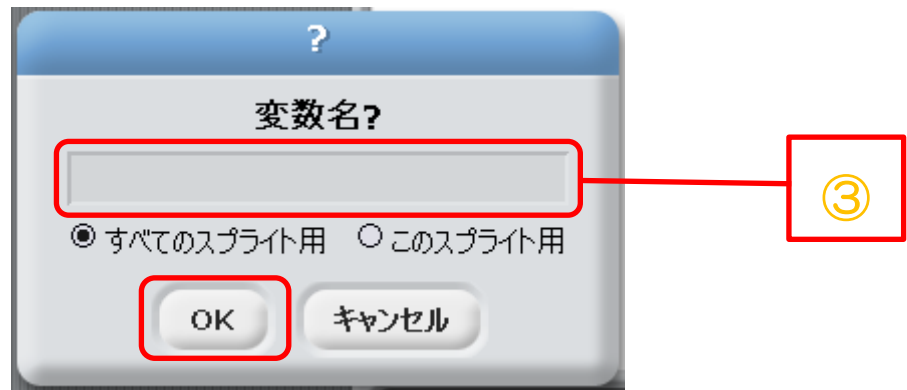


図 2 変数をつくる②



図 3 変数をつくる③

カウントダウンタイマーをつくる

- ① **変数をつくる**の①～③で変数「Timer」を作成します。
- ② 作成された変数のブロックとコマンドグループパレット「制御」のブロックを図 2 のように組み合わせると、カウントダウンタイマーを作成することができます。
- ③ カウントダウンを始めたい数字（秒数）を入力します。（この場合 60 秒とします）
- ④ 『Timer を〇ずつ変える』のブロックに変化させたい間隔の数字を入力します。通常は『1』です。これを『-1』に変更することでカウントダウンが可能になります。逆に『1』のままだと『60』から 1 ずつカウントアップされます。

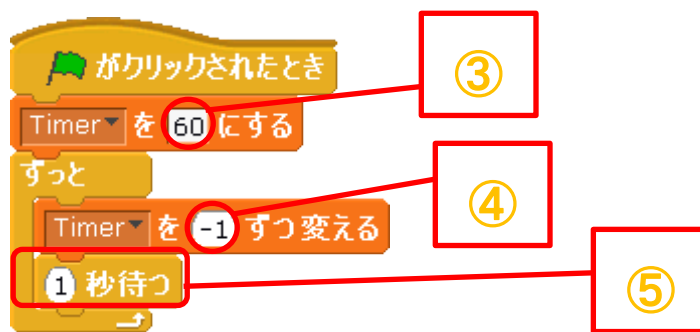


図 4 カウントダウンタイマーをつくる

- ⑤ 赤で囲まれた『1 秒待つ』というブロックを入れると、1 秒ずつカウントダウンされるようになります。これで 1 分間のカウントダウンタイマーができます。

点数表示のしくみをつくる

- バナナに触れたらスコアを 1 ずつ増やすプログラム、によって説明します。

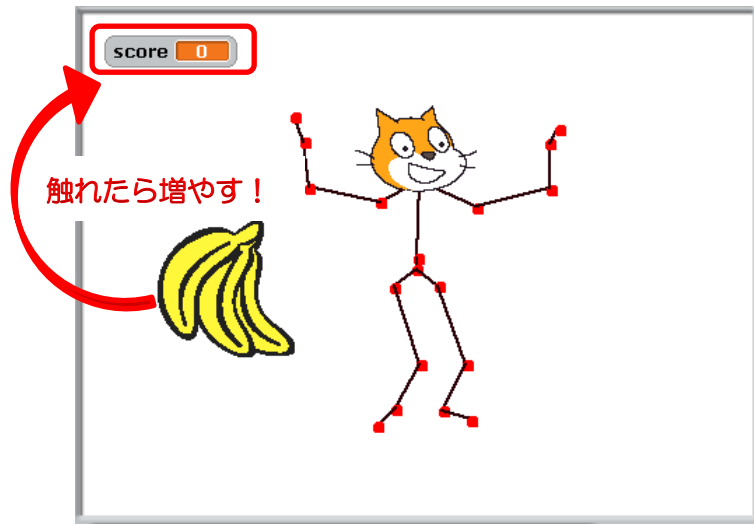


図 5 点数表示のしくみをつくる①

- バナナのスプリクト

- ① **変数をつくる**の①～③で変数「score」を作成します。
 - ② 作成された変数のブロックとコマンドグループパレット「制御」、「動き」、「調べる」の各ブロックを図 4 のように組み合わせます。
 - ③ score を『0』にします。ここに入れたものスタート時点の数字となります。
 - ④ 『口に触れたら』のブロックの口の部分を赤色（スケルトンの関節の色）にします。
 - ⑤ score を 1 ずつ変えるとします。（一度触れるたびに 1 ずつ増えます。）
- 以上で、点数を表示するプログラムの完成です。



図 6 点数表示のしくみをつくる②

流れる風景をつくる

図5のように、風景が流れていくように見えるプログラムの作り方です。



図7 流れる風景の例

Scratchの座標は図6のようになっています。

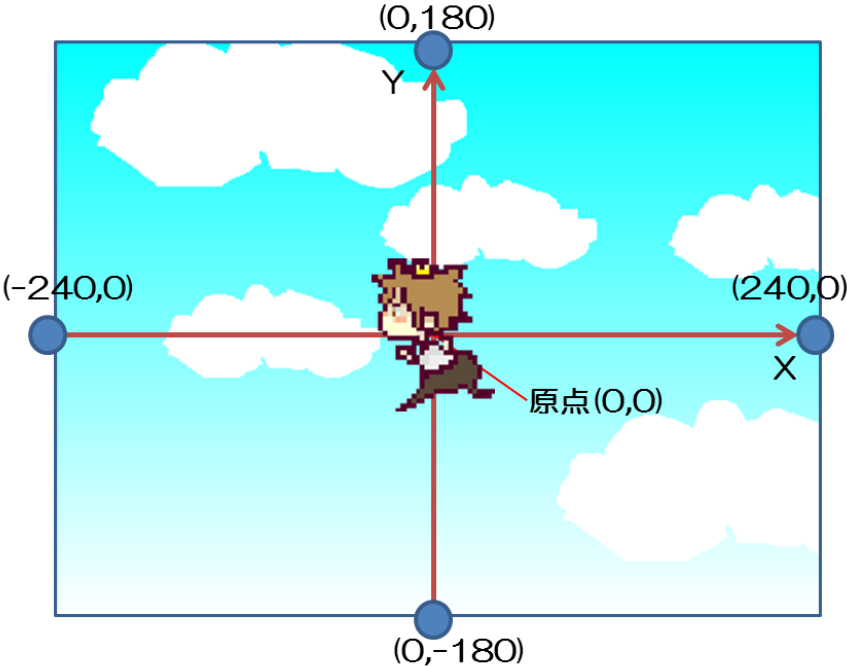


図8 ステージの座標

<次のページへつづきます>

・動くもの（図7では「雲」のSprite）のScript

- ① コマンドグループパレット「制御」のブロックと「動き」のブロックと「演算」のブロックを図5のように組み合わせます。
- ② 動いていく分の座標を入力します。（この場合5ずつ変わっていきます）
この場合「x座標」を変えて横に動くようにしていますが「y座標」にすると縦に動かすこともできます。
- ③ 折り返したい座標（この場合画面の一番端（座標250））にSpriteが流れると
- ④ 戻りたい位置に戻るように値を入れる（この場合元の位置に戻すので-240）



図9 流れる風景をつくる

実行の条件をつくる

- ・「実行（緑の旗）がクリックされた時」以外のタイミングでプログラムを実行したい場合

例えばサンプルゲームの「Kinect Ball Barance Game」（ピンボール）ではゲームが起動するタイミングが「実行（緑の旗）がクリックされた時」ではなく、「スクリプト（棒）が画面の端に触れたら」ボールが動くようにプログラムが作成してあります。

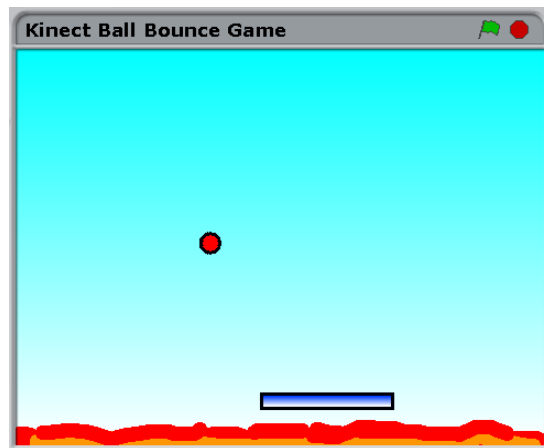


図 10 実行の条件をつくる①

- ① コマンドグループパレットの「制御」をクリックします。
- ② ブロックパレットの「～を送る」のブロックの▼をクリックすると「新規…」と表示されるので「新規…」をクリックします。
- ③ メッセージを入力します。（自分が実行したいプログラムに関連した言葉にしたらプログラムを作成するときにわかりやすいです。例：プログラムの実行の時→game_on）
- ④ メッセージを送りたいタイミングの場所に「～を送る」のブロックを組み込みます。（「～を送る」のブロックの▼をクリックするとメッセージが選択できます。）
- ⑤ コマンドグループパレットの「制御」をクリックします。
- ⑥ ブロックパレットの「～を受け取った時」の下に、メッセージを受け取った時に実行したいプログラムを作成しましょう。

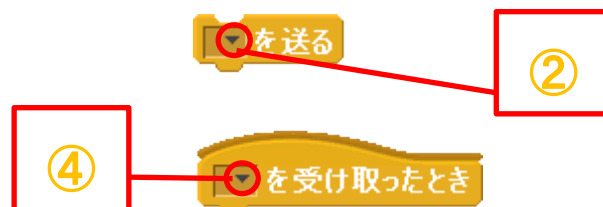


図 11 実行の条件を作る②

例：サンプルゲームの Kinect Ball Barance Game の場合

図 7 は棒のスク립トの一部、図 8 はボールのスク립トの一部です。

※「Kinect Ball Barance Game」ではゲームを起動時には「game_on」という変数が「0」、起動時には「1」になるように設定してあります。

- ⑦ 図 7 の棒のスク립トを見ると、「もしボールが端に触れたなら」「game on を送る」とブロックが作成してあります。



図 12 実行の条件をつくる③

- ⑧ 図 8 のボールのスク립トを見ると最初が「game on を受け取った時」になっています。つまり「game on を受け取った時」から下のブロックは「棒が端に触れて game on が送られた」時に初めて実行され、ボールが動くように設定してあります。

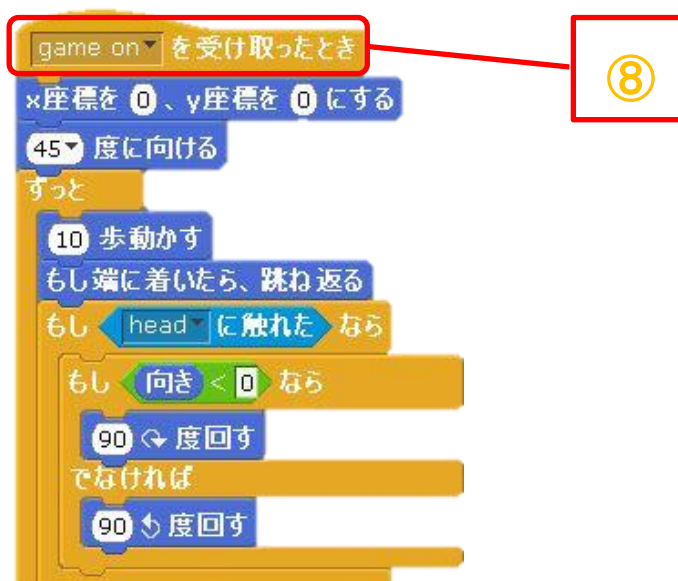


図 13 実行の条件をつくる④

スケルトンセンサーの値見取り図

スケルトンのセンサーはこのように対応しています。

