

(2) 変数, 条件式, 繰り返し, 条件分岐

Scratch

変数

- 一時的にデータを入れておく “箱”
- Scratchにおける変数の種類
 - グローバル変数
 - どのスプライトからでも利用できる
 - ローカル変数
 - 特定のスプライトに所属
 - 所有するスプライトしか変更できない
 - 他のスプライトは参照（見ること）しかできない
- 出来る限りローカル変数を利用

変数の作成



すべてのスプライト用＝グローバル変数
このスプライト用＝ローカル変数

変数の操作



変数に値を入れる

変数の値を変更する（加える）

ステージへの表示・非表示

歩数をカウントさせる

- 下のようなスクリプトを作成



- 実行すると「歩数」変数の値が変化する

変数の使える場所

- 基本的には丸空白に使用できる

こんなところ

- 条件式の四角空白も使用可

こんなところ



1 度に移動する幅を変更する

1. 「歩幅」という変数を追加
2. 右のスク립トを作成
3. 歩幅を変えて動かしてみる



条件式

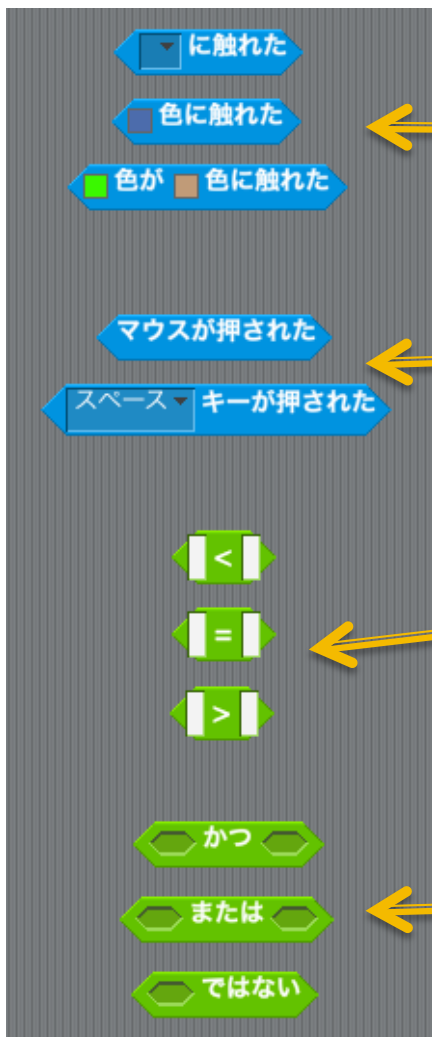
- 条件を判断する
 - 結果は「成立」か「不成立」
- Scratchでは六角形をした部品



こんな感じの部品

- 他の部品と組み合わせて利用
 - 組み合わせ可能な箇所は六角形の穴がある

Scratchの主な条件式



← スプライト同士などの接触判定

← キーボードやマウス入力のイベント判定

← 数値の大小関係の判定

← 複数条件 (AND, OR) と否定

条件式を使う命令

もし なら、ずっと

まで繰り返す

もし なら

もし なら

でなければ

まで待つ

繰り返し

- 条件が成立する間繰り返す（上）
- 条件が成立するまで繰り返す（下）

条件分岐

- 条件が成立する時のみ内側を実行（上）
- 条件によって実行する命令を変更（下）

条件が成立するまで待つ

- 他のスプライトと同期する場合等に利用

百歩以上歩く

- 以下のようにスクリプトを変更してみる



ここを変更

条件分岐を使って見る



ここを追加

考えてみよう

- 「歩幅」がどんな数でも疲れるようにする
 - 今は「歩幅」が100の約数でないとダメ

他のスプライトとの相互作用

1. 適当なスプライトをステージに配置
※ここでは新しいスプライトの名前を「バナナ」に変更している
2. 以下のようにスクリプトを変更
(けっこう大幅な変更なので一から作っても良い)

