

# Scratchを用いた 非言語プログラミング

変数・条件式・繰り返し・条件分岐  
キーボード・マウス・イベント

# 変数

- データを入れておくための箱
  - 箱には名前がつけられる
- 変数の種類
  - 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）

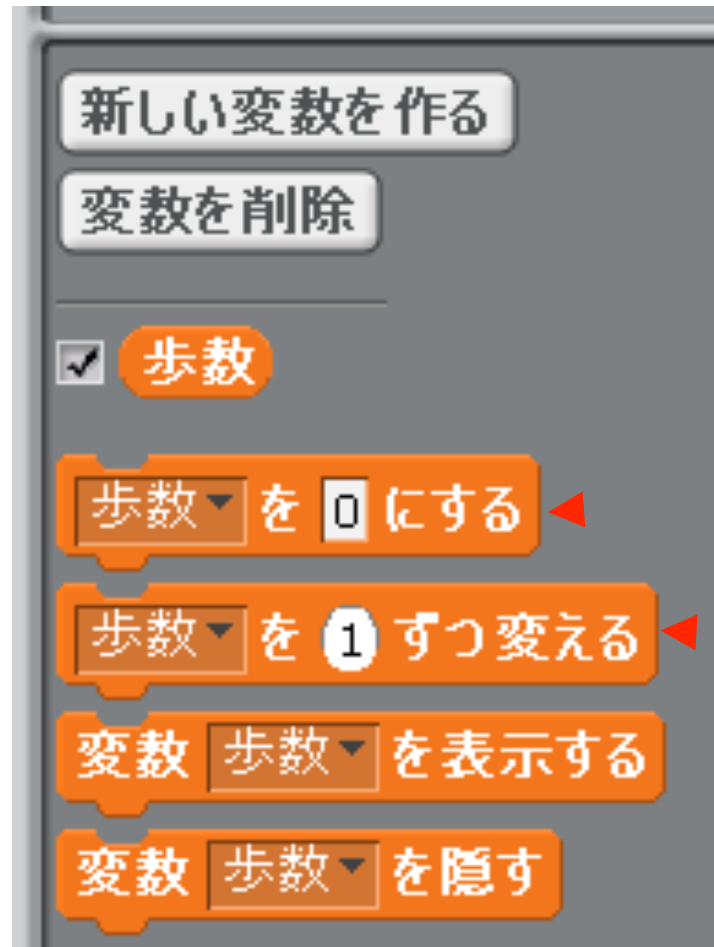
# 変数の作成



このスプライト(猫)に関する  
データを入れておくための  
「歩数」という名前の箱を作る

# 変数の操作

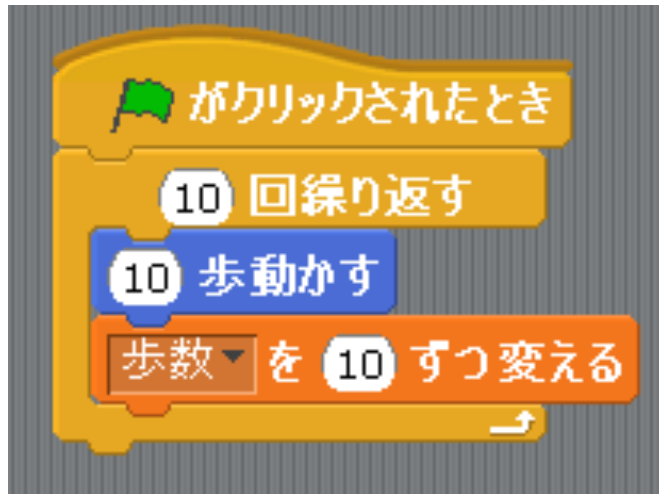
## (箱の中身を書き換える)



- ・ 変数に値を入れる
- ・ 変数の値を変える(加える)

# 変数の利用例1: データを覚えておく

- 以下のスクリプトを作成

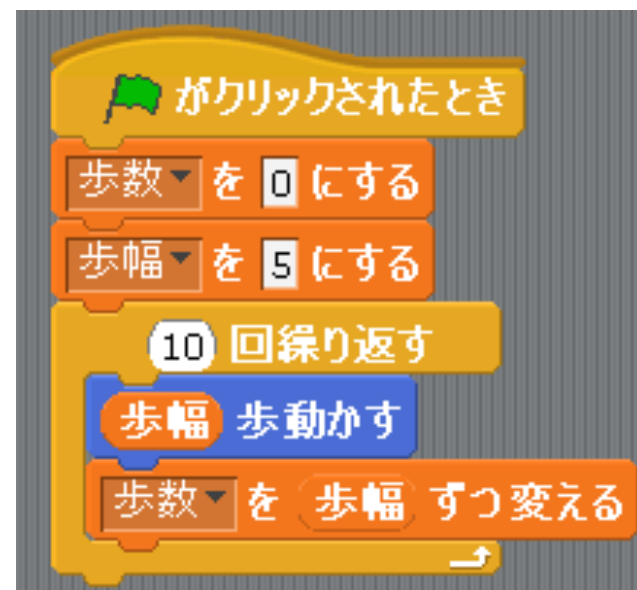


- 実行すると「歩数」の値が変化する  
– ステージに表示したくないとき



# 変数の利用例2: 同じデータを複数の場所で使う

- 右のスクリプトを作成
- 変数「歩幅」を作成
  - ドラッグアンドドロップで  
数字のところに入れる



# コスチューム

- 1つのスプライトの別の姿

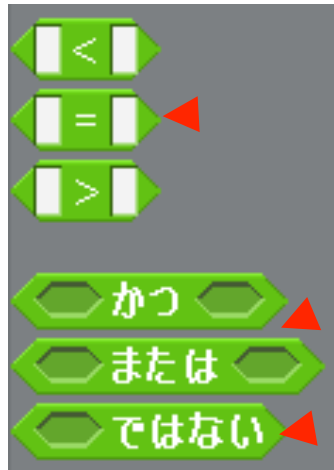
前回の「スプライトを作成する」と同じように  
コスチュームを作成できる

# 条件式

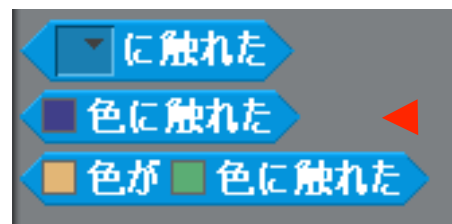
- 条件を判断する
  - 結果は「成立」か「不成立」
  - Scratchでは六角形の部品
- 他の部品と組み合わせて使う
  - 六角形の穴に入れる



# Scratchの主な条件式



- ・数値の大小関係の判
- ・複数条件
- ・条件の否定



- ・接触判定

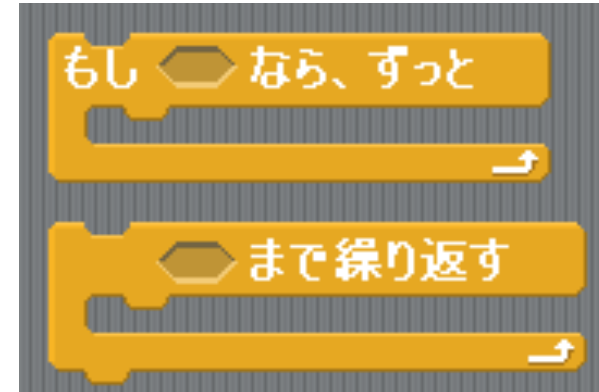
- ・変数を使うことが多い 例:



# 条件式と組み合わせて使う命令

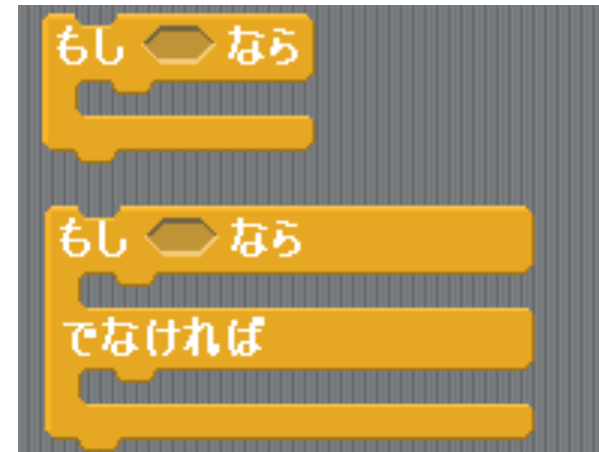
- 繰り返し

- 上:条件が成立する間繰り返す
- 下:条件が成立するまで繰り返す



- 条件分岐

- 上:条件が成立するときだけ実行
- 下:条件が成立するときとしないときで違う命令を実行



# 繰り返しの利用例

- 以下のスクリプトに変更



繰り返す回数ではなく  
繰り返す条件を指定

# 条件分岐の利用例

- 以下のスクリプトに変更

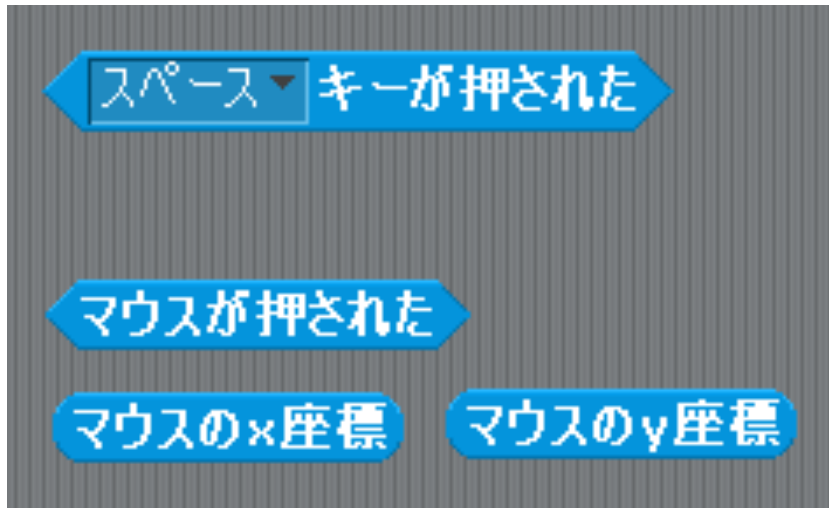


- この条件が満たされる時だけ
- 中身の命令が実行される

# キーボードとマウス

## 調べる

- その時の状態を調べる



## イベント

- 押されたときに反応する



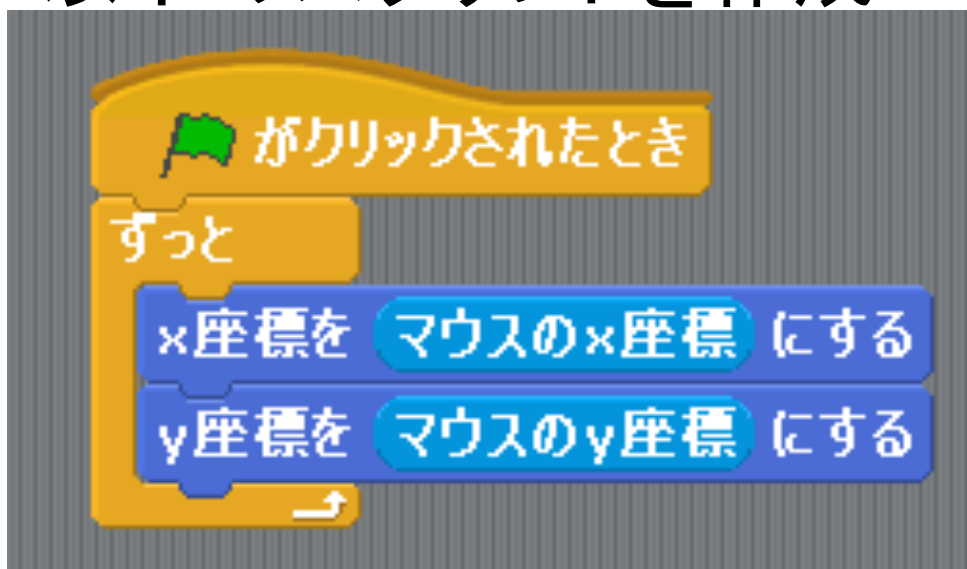
# キーの状態を調べる

- 以下のスクリプトを作成



# マウスの座標を調べる

- 以下のスクリプトを作成



- X座標: ステージ上で左右方向の位置
- Y座標: ステージ上で上下方向の位置

# 応用: マウスを追いかける猫

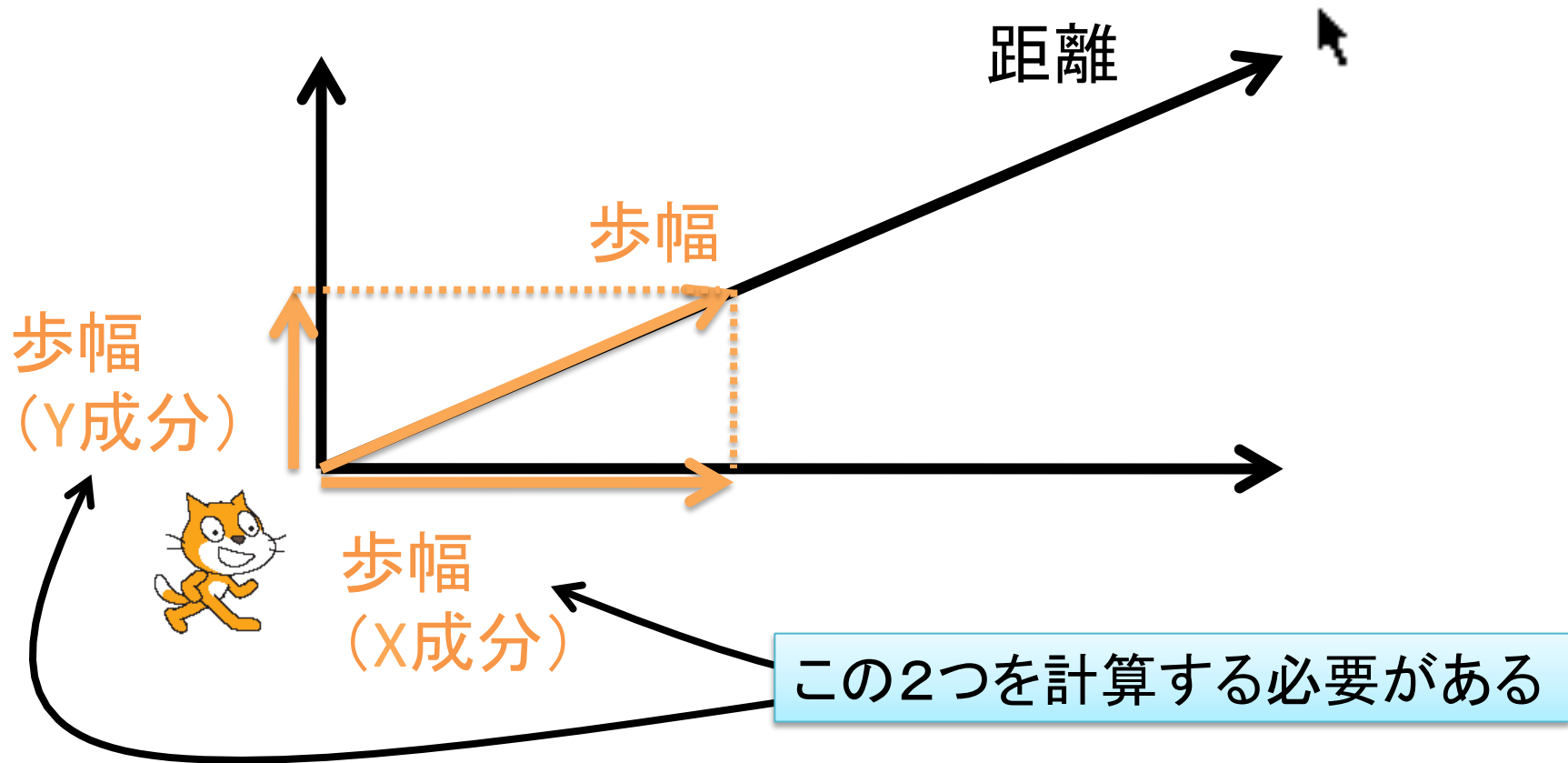


7

歩幅分だけ進む



# マウスを追いかける方法



# マウスを追いかける方法(つづき)

1. 割合 =  $\frac{\text{歩幅}}{\text{マウスのポインターまでの距離}}$

2. 距離(X) =  $\text{マウスのx座標} - \text{x座標}$

3. 歩幅(X) = 距離(X) × 割合

- Y成分も同様
- Scratchでは距離が簡単に調べられるが、距離(X)と距離(Y)から計算することもできる

# マウスを追いかける猫のスク립ト

- 以下のスク립トを作成



The image shows a Scratch script for a cat following the mouse. The script is as follows:

- がクリックされたとき** (When clicked)
- 歩幅** を **5** にする (Set step size to 5)
- ずっと** (Forever loop)
  - x座標を**  $\text{マウスのx座標} - \text{x座標} * \text{歩幅} / \text{マウスのポインターまでの距離}$  ずつ変える (Change x coordinate by  $\text{マウスのx座標} - \text{x座標} * \text{歩幅} / \text{マウスのポインターまでの距離}$ )
  - y座標を**  $\text{マウスのy座標} - \text{y座標} * \text{歩幅} / \text{マウスのポインターまでの距離}$  ずつ変える (Change y coordinate by  $\text{マウスのy座標} - \text{y座標} * \text{歩幅} / \text{マウスのポインターまでの距離}$ )

# イベント

- キーボード
  - 何かキーが押された
- マウス
  - マウスボタンが押された
  - マウスが動いた
- タイマー
  - ある時刻になった
  - ある時間が経過した

# イベントの例

- 右のようなスクリプトを作成
  - 「複製」機能を使うと楽
- 上下左右の矢印を押したとき
  - ステージでは何が起きる？
  - スクリプト画面では何が起きる？
    - 実行されているスクリプトが光る

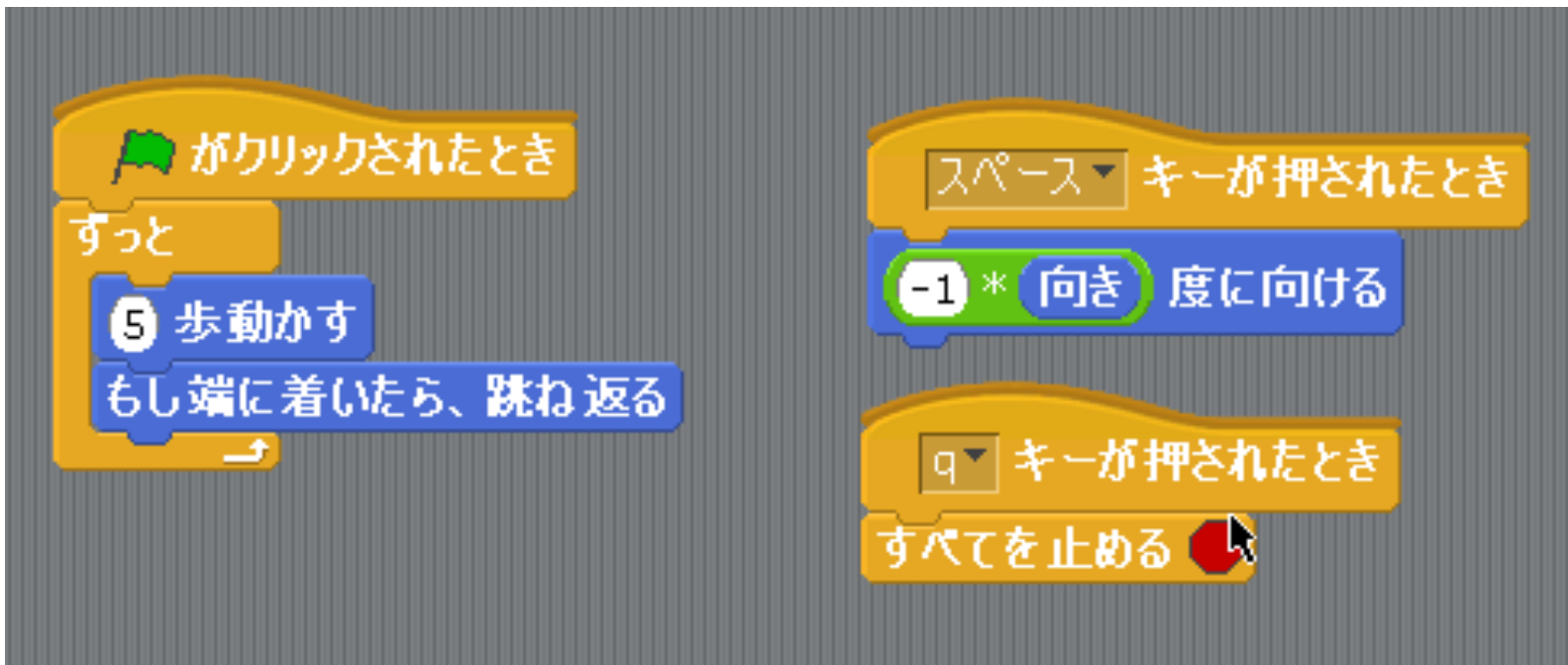


# イベント割り込み

- スクリプトを実行している最中にイベントが発生した時に別のスクリプトを割り込んで実行

# キーボードで割りこむ

- 以下のスクリプトを作成



The image shows a Scratch script with three event triggers and their corresponding actions:

- Click event:** When the green flag is clicked, a loop block labeled "ずっと" (Forever) contains:
  - Step 5: Move forward (歩動かす)
  - Jump back if at the edge (もし端に着いたら、跳ね返る)
- Space key event:** When the space key is pressed, turn -1 degrees in the direction of "向き" (Direction).
- Q key event:** When the q key is pressed, stop everything (すべてを止める).

# クリックで割りこむ

- 右のようなスクリプトを作成

The image displays three Scratch scripts on a dark grey background. The top-left script is for a character's movement logic. It starts with a 'when clicked' event, followed by 'set thinking to 0'. A 'when thinking is 0' loop contains 'move 5 steps' and 'if at edge, bounce back'. The top-right script shows two key events: 'space key pressed' which turns the character -1 degrees, and 'q key pressed' which stops all effects. The bottom script is for 'sprite 1' and uses a 'when clicked' event. It has a 'when thinking is 1' loop with 'set thinking to 0' and 'say hello for 2 seconds'. Outside the loop, it has a 'when thinking is 1' loop with 'set thinking to 1' and 'say um... for 2 seconds'.

```
when clicked
  set thinking to 0
  loop when thinking is 0
    move 5 steps
    if at edge, bounce back

space key pressed
  turn -1 degrees

q key pressed
  stop all effects

sprite 1 clicked
  loop when thinking is 1
    set thinking to 0
    say こんにちは! for 2 seconds
  when thinking is 1
    set thinking to 1
    say うーん... for 2 seconds
```