

前回のおさらい

- ・繰り返し処理
- ・変数、乱数

(・スプライトを隠す・表示する)

これを使って、クリックゲームの作成

今回の新しい内容

- ・条件分岐 (ほかのプログラム言語等では IF 文とか Excel だと IF 関数とか)

条件に当てはまるか、当てはまらないかで処理の内容を変えること。

条件分岐というだけあって、条件を必ず必要とする。

Scratch の条件は、六角形の穴を埋めていく。

<基本パターン 1>

条件に当てはまった時に、処理を行う。

(条件に当てはまらなかったら、処理を行わない)



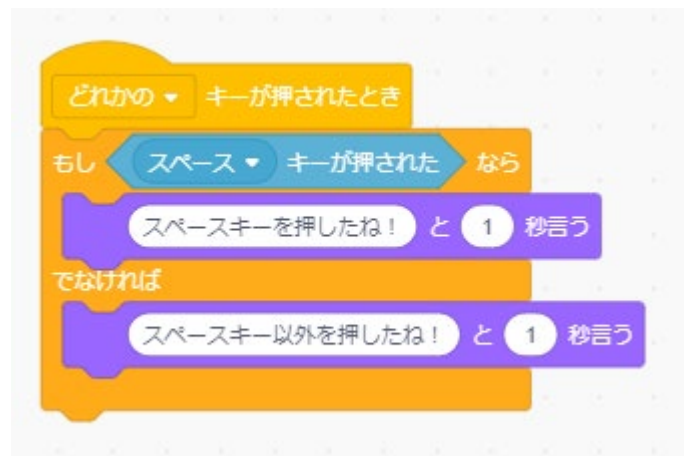
<基本パターン 2>

条件に当てはまった時に、処理を行う。

条件に当てはまらなかったら、別の処理を行う。



実際には、このような形で使う。



でも、これだとなんかつまらない…。

先日のクリックゲームを少し改良してみます。



### 修正ポイント

※押された数=10 まで繰り返す

これ、マウス連打してしまうと、一気に 10 を超えてゲーム終了しなくなります。

(単にこのクリックゲームのバグ (不具合) になります、このバグを取り除く作業をデバッグと言います)

→ 押された数>9 に変更する

### 改良ポイント

※この押された数>9 まで繰り返す→押された数>19 まで繰り返す

押された数 (スコア) が 10 を超えたら、少し難易度を上げてみましょう。

※難易度を変更するために、変数「秒数」を追加します

※初期処理として、秒数を 1 にする

※1 秒待つ、のところを「秒数」秒待つ、に変更

**※もし、押された数>9 なら、秒数を 0.7 にする**

※最後の 10 回捕まえたよ！を 20 回捕まえたよ！に変更

これだけの変更 (工夫) で、ぐっと難易度が変わってきますよ。

(変更後の例は次頁)

構造化プログラミングの、順次処理 (逐次処理)、繰り返し処理、分岐処理をこれで網羅しました。

次回講義はこの条件分岐処理をもう少し複雑にやってみようと思っています。

(複数条件がある場合等、また、Excel の IF 関数を利用した例)



今回コメントをつけてみました。

※コメントについて

プログラム自体にメモ書きを残すことが出来る機能。

これを残しておくことで、プログラム作成時にどのような考え方で作成したかなど、ヒントを残すことが出来る。久々にプログラム触ろうとしたときに早く思い出せる等、メリットが多いので基本的を書くようにするとよい。

動きを確認したら、保存しておくこと。

その後、ファイル→新規を選び、新しい画面にしておく。

## ・配列について

Scratch をはじめ、いろいろなプログラミング言語で、配列というものが使えます。  
変数は1つしか値を持つことが出来ませんが、配列は複数個持つことが出来ます。  
三段ボックスみたいなイメージでいいかと思います。  
Scratch では配列はリスト、という名称になっています。  
果物、というリストを試しに作成したとします。

りんご	果物、というリストの1番目
バナナ	果物、というリストの2番目
みかん	果物、というリストの3番目

配列（リスト） 果物

## 作成方法



この後、リストの名称を入力する画面が出るが、変数とほぼ同じなので割愛。  
(次頁に続く)



このようにリストに関する項目が追加される。  
作成された時点で、リストは空である。

果物の中にデータを追加する場合、  
「なにかを果物に追加する」（「なにか」は変更  
すること）を使用する。

←ぐらいに、果物の1番目になにかを、と、  
果物の1番目になにかで、とあるが…



の違いです。

具体的な例（スプライトにじゃんけんをさせてみよう）

リスト「じゃんけん」を作成する。

初期処理として、グー、チョキ、パーをリスト「じゃんけん」に入れる。



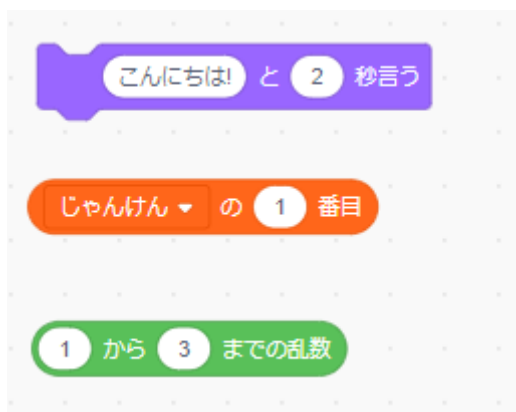
最初にすべてを削除しておかないと、実行するたびに「グー、チョキ、パー」が追加されてしまうので注意する。（試しに動かすと良いかと思います）

その後、スプライトにじゃんけんの結果を言ってもらう。

乱数で、1～3までの数を発生させる。

1～3までの数字が出来たら、リストの○番目の内容を話してもらう。

1が発生したら、グーと、2が発生したら、チョキと、3が発生したらパーと話す。



この3つのブロックを組み合わせてみよう！

（解答例が次頁にありますが、挑戦してみよう）

<解答例>



※厳密に言えば、配列とリストは違うものになります。  
配列は最初に「〇〇段ボックス」と決めておかないといけない、という決まりがあります。  
リストは「〇〇段ボックス」の何段かを変更できます。  
Scratch はリストだけ扱うことが出来ます、が、配列と今は同じイメージでよいでしょう。

この配列（リスト）を利用して、様々なアルゴリズムが作られている。  
例えばテスト等の順位をつけるアルゴリズムも作成できる。