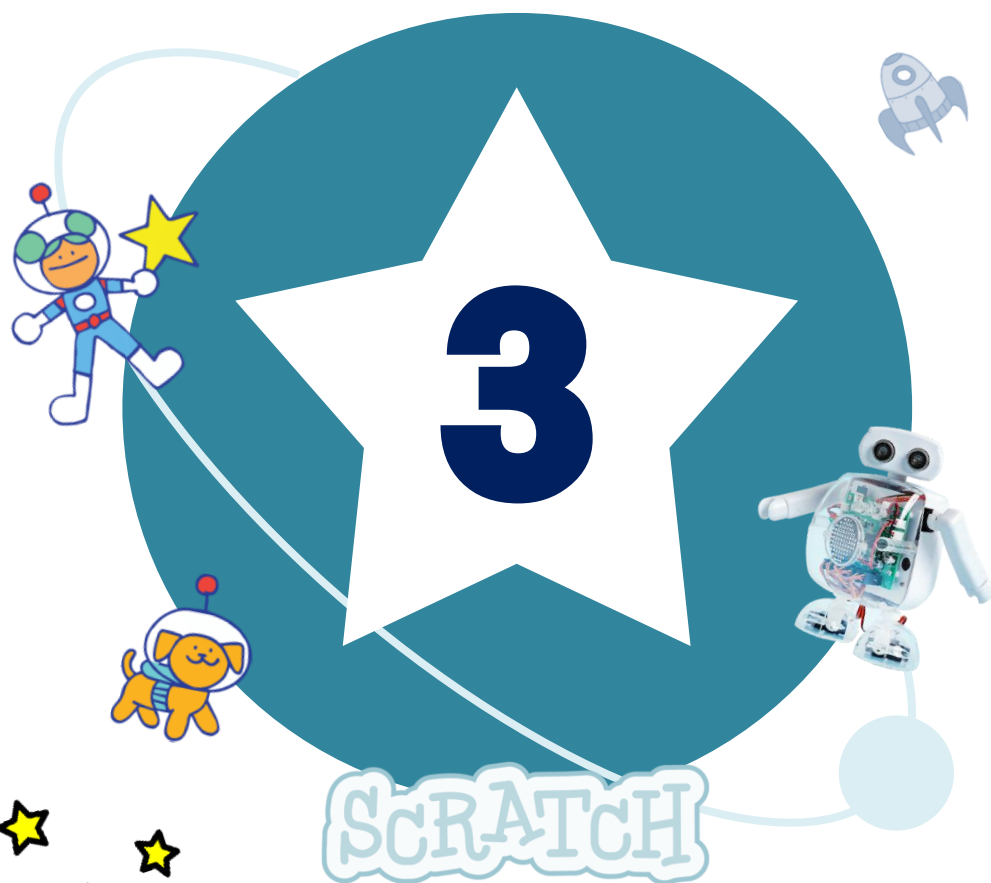


# スクラッチであそぼう



# 条件分岐

今回は『もし～なら』という条件分岐のブロックについて学んでいきます。

プログラムの流れを変えるブロックは、ループ(繰り返し)ブロックの他に、**条件分岐**という考え方があります。「分岐」は**分かれる**という意味です。

プログラムの流れは、上から下に順番に実行されています。

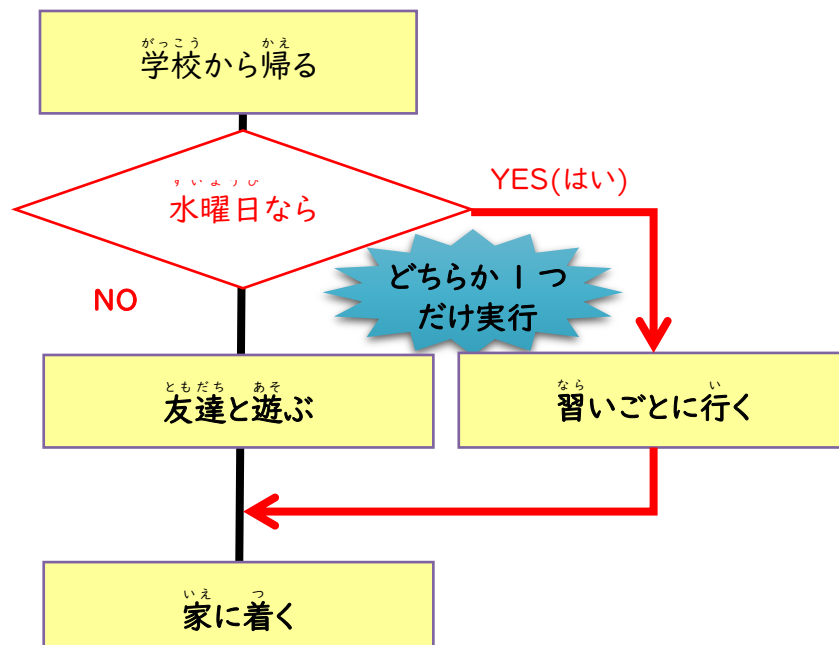
この流れを、条件によって2つの道や3つの道に枝分かれさせることができます。

このときに使うのが**条件分岐ブロック**です。

## 💡 条件分岐の考え方

もし <...> の場合は、XXX する

例) もし水曜日なら、習いごとに行く



# 条件分岐ブロックの使い方

## ◆スクラッチの条件分岐ブロック

	に入れた条件を満たしていると、ブロック内のスクリプトを実行する。満たしていなければ、プログラムは下に進む。
	に入れた条件を満たしていると、ブロック内のスクリプトを実行する。満たしていなければ、【でなければ】の中のスクリプトが実行してから、下に進む。
	に入れた条件を満たすまで、スクリプトの実行を待つ。条件を満たすと、プログラムは下に進む。

## ★やってみよう!★

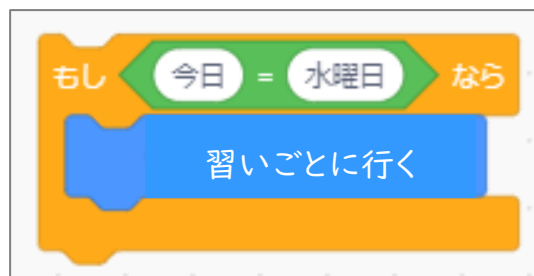
### 1. 【もし<>なら】ブロック

条件に一致したなら、ブロックの中のスクリプトを実行する。一致しなければ、プログラムは下に進み、このブロックの下にある処理が実行される。

条件(もし~なら)



水曜日の場合、習いごとの後に家に着く



YES の場合のみ実行

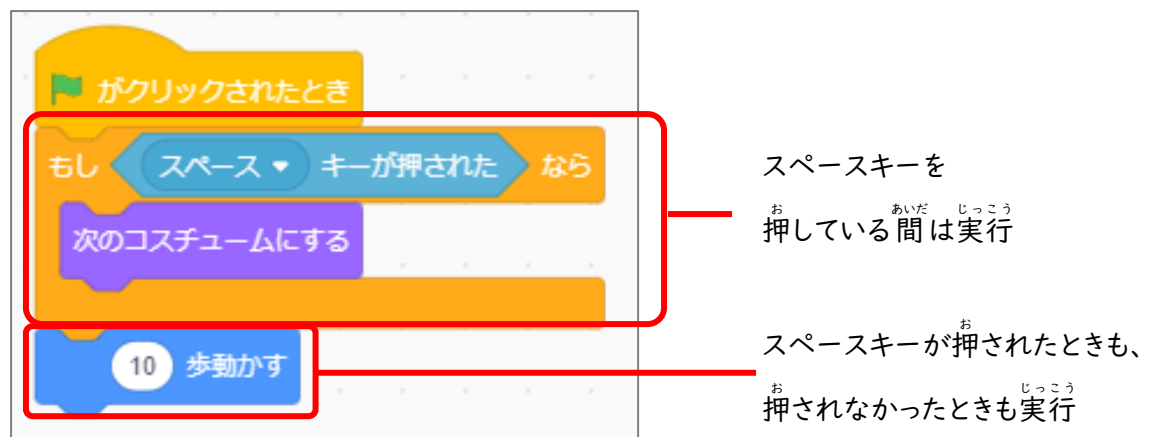
水曜日以外の場合、習いごとは行かずに家に着く



もしスペースキーが押されたなら、スクラッチキャットのコスチュームが変わるプログラムを作成しよう。

スペースキーが押されなかった場合は、【つぎのコスチュームにする】は実行されず、下の【10歩動かす】のみが実行されるよ。

この流れの考え方が基本なので、よく理解しておいてね！

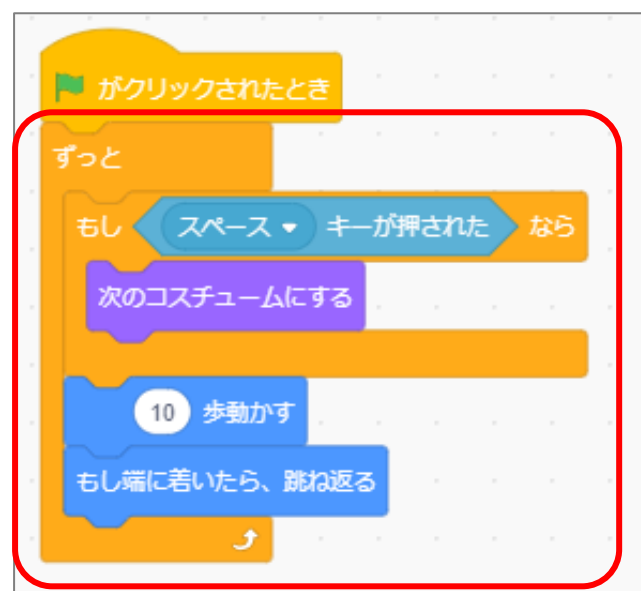


💡スペースキーが押されたときの動きを確認するには、先に押しながらスタートさせるとわかりやすいよ。

## ★やってみよう!★

1のプログラムを変更して、【ずっと】繰り返すブロックで囲んでみよう。

ずっと動き続け、スペースキーが押されたときにコスチュームが変更されるプログラムになることを確認してね。



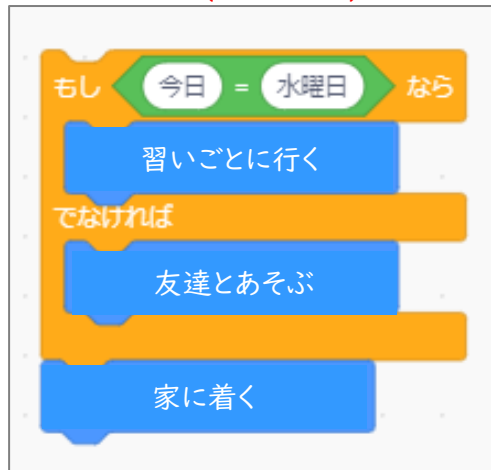
## 2. 【もし<>なら でなければ】ブロック

条件に一致したなら、ブロックの中のスクリプトを実行する。

一致しなければ、【でなければ】の中のスクリプトが実行される。

どちらの場合もこのブロックの下にある処理は実行される。

条件(もし~なら)



YES の場合のみ実行

NO の場合のみ実行

かならず実行

1 のプログラムを変更して、スペースキーが押されなかったときの動きを

【でなければ】に入れてみよう。1 とは動き方が変わるので、よく理解しておいてね!

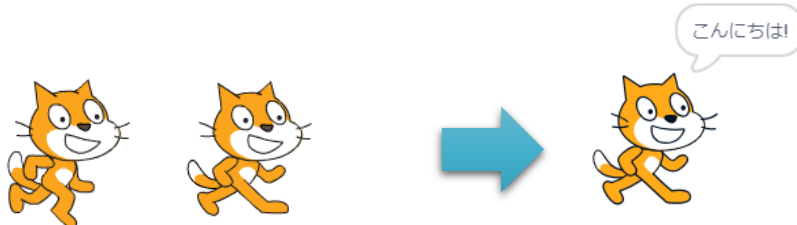


スペースキーが押されなかったとき

## ★問題★

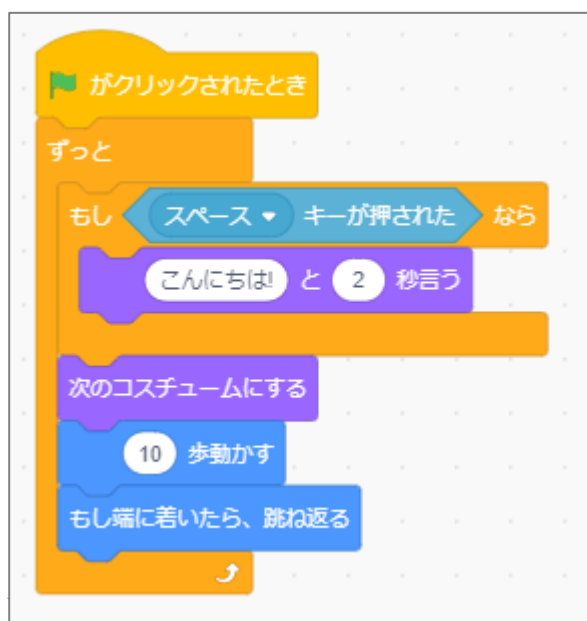
ずっとコスチュームを変えながら動き続けているスクラッチキャットが、スペースキーを押すと止まって『こんにちは!』と2秒間あいさつするプログラムを作ってみよう!

あいさつは、『見た目』グループの  ブロックを使ってね。



スペースキーを押すとあいさつする

## ★答え★



【ずっと】の中に【もし】を入れると、  
いつスペースキーが押されても  
『こんにちは!』と言うようにできるよ

## ★問題★

好きなスプライトを追加して、次の動きになるようにプログラムを作成しよう。

- ・スタートされたとき『スタート!』と2秒いう。
- ・ずっと右方向に、コスチュームを変えながら歩き続ける。
- ・もし端についたら『ゴール!』と2秒いう。

💡 ヒント : 『調べる』グループの  ブロックで、端にふれたかどうかを調べられるよ!

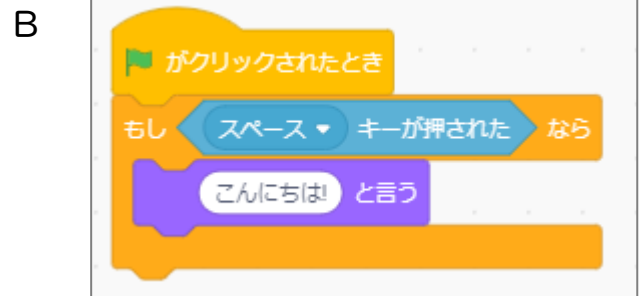
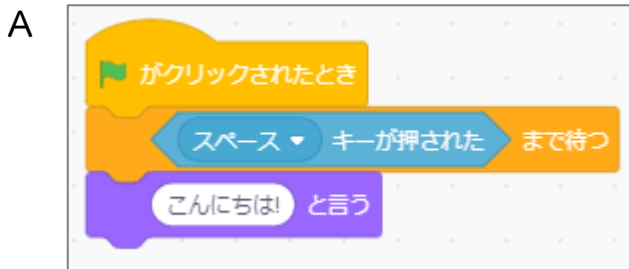
### 3. 【<>までまつ】ブロック

<>で指定した条件に合うまで、スクリプトの実行を待つ。<>が起きると、下にプログラムが進む。



次の2つのプログラムを作成して、動きの違いを確認しよう。

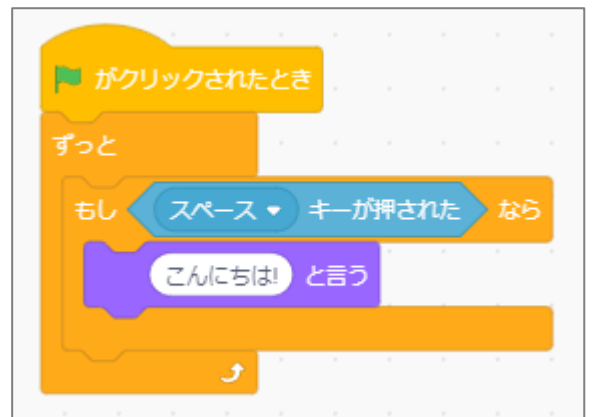
Aの【<>までまつ】はプログラムを停止して待ち状態になるけれど、Bの【もし<>なら】はブロックが実行されたときに条件が合っていないと通り過ぎてしまうよ。



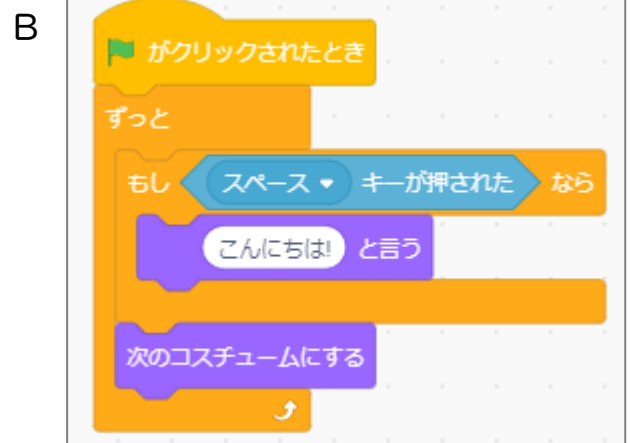
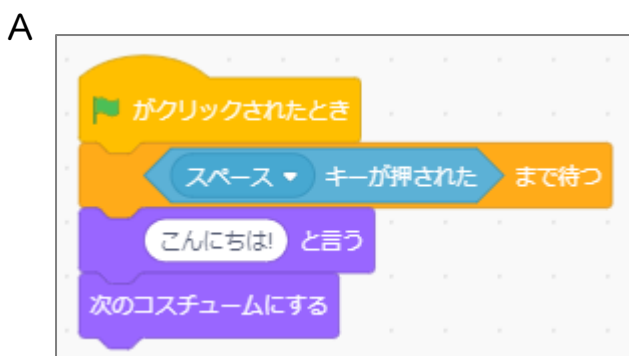
では、Bのプログラムに【ずっと】を加えて変更してみよう。

これで動きはAもBも同じになったね。

けれど、実は大きな違いがある！



次のようにプログラムを変更して、どんな違いがあるのか考えてみよう！



気が付いたかな。

【ずっと】はとても便利な繰り返しブロックだけれども、下に他のブロックをつなげられない。

連結用のでっぱりが無いよね。プログラムの流れを考えて、上手に使うようにしよう。



## れんしゅうもんだい 練習問題

次のプログラムのうち、「スペースキーが押されたときは『スペースキー』といい、a キーが押されたときは『a キー』といい、どのキーも押されなかったときは『こんにちは!』という」プログラムはどちらかな。作成して動きを確認しよう。

A



B



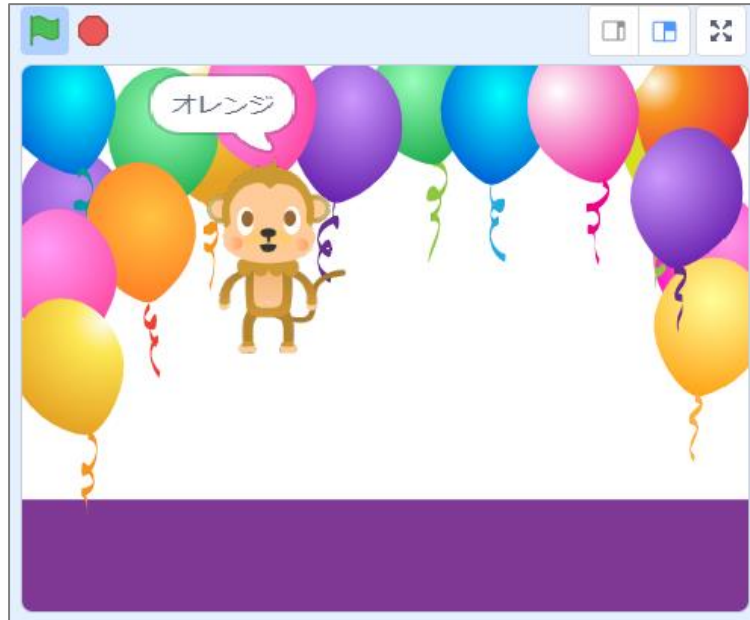
💡 ヒント: プログラムは上から下に実行されるよ。

答えは最後のページの右下にあるよ。



## かくにんもんだい 確認問題

『3\_<sup>かくにんもんだい</sup>確認問題.sb3』ファイルを開き、次の動きになるようにプログラムを<sup>かんせい</sup>完成させよう。




1. キーボードの<sup>じょうげさゆう</sup>上下左右の矢印キーが押されたとき、コスチュームを<sup>か</sup>変えながらその<sup>ほうこう</sup>方向に<sup>ある</sup>歩くように<sup>せってい</sup>設定しよう。

💡 <sup>うわむ</sup>上向き矢印キーの部分は<sup>ぶぶん</sup>完成しているので、<sup>さんこう</sup>参考にしてね。

2. スプライトが<sup>ふうせん</sup>風船にふれたとき、<sup>いろ</sup>色の<sup>なまえ</sup>名前を<sup>い</sup>言うように<sup>せってい</sup>設定しよう。

💡 <sup>むらさきいろ</sup>紫色の部分は<sup>ぶぶん</sup>完成しているので、<sup>さんこう</sup>参考にしてね。

3. <sup>ふうせん</sup>スプライトが風船にふれていなときは『<sup>はな</sup>こんにちは!』と<sup>せってい</sup>話すように設定しよう。

💡  <sup>ばしょ</sup>ブロックを、ある場所に 1 つ追加するだけで<sup>かんせい</sup>完成するよ!

プログラムは、<sup>うえ</sup>上から<sup>した</sup>下に<sup>なが</sup>流れて<sup>じっこう</sup>実行されることを思い出してね!

## スプライト『Monkey2』

