

SCRATCHプログラミング

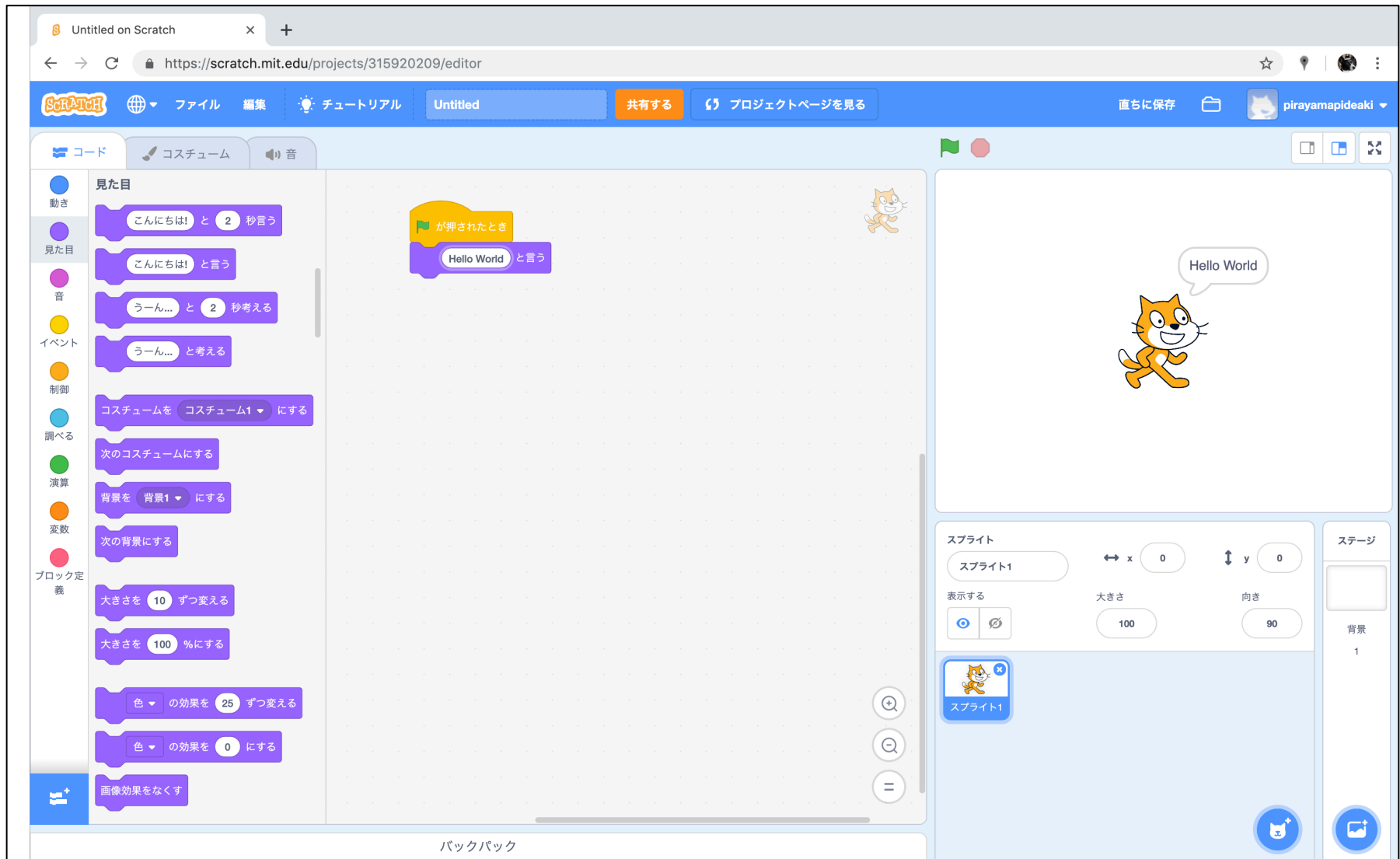
Vol. 3

目白大学メディア学部メディア学科
平山秀昭

- ・プログラミングの勉強のスタートは、
「Hello World」と書くプログラムから始まります。
こんな感じです。

```
main()  
{  
    printf("Hello World");  
}
```

・SCRATCHのHello Worldのプログラムは、こんな感じ。



- ・プログラミングの勉強では次に「変数の宣言文」と「代入文」というものを学びます。
- ・変数の宣言文とは、以下のような文です。

```
int a;
```

これは、「aという変数を使うことを宣言する」という意味です。

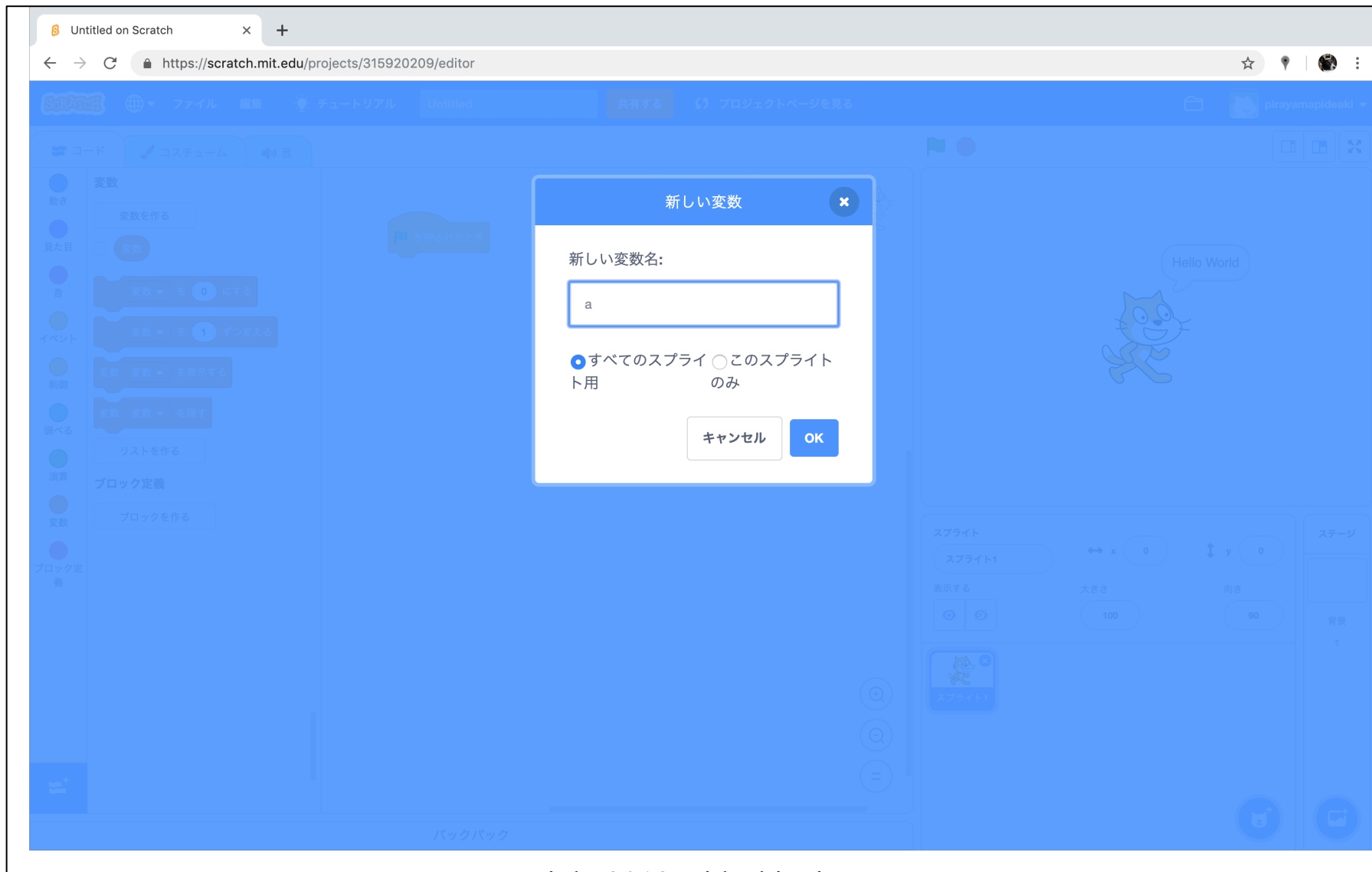
- ・代入文とは、以下のような文です。

```
a = 10;
```

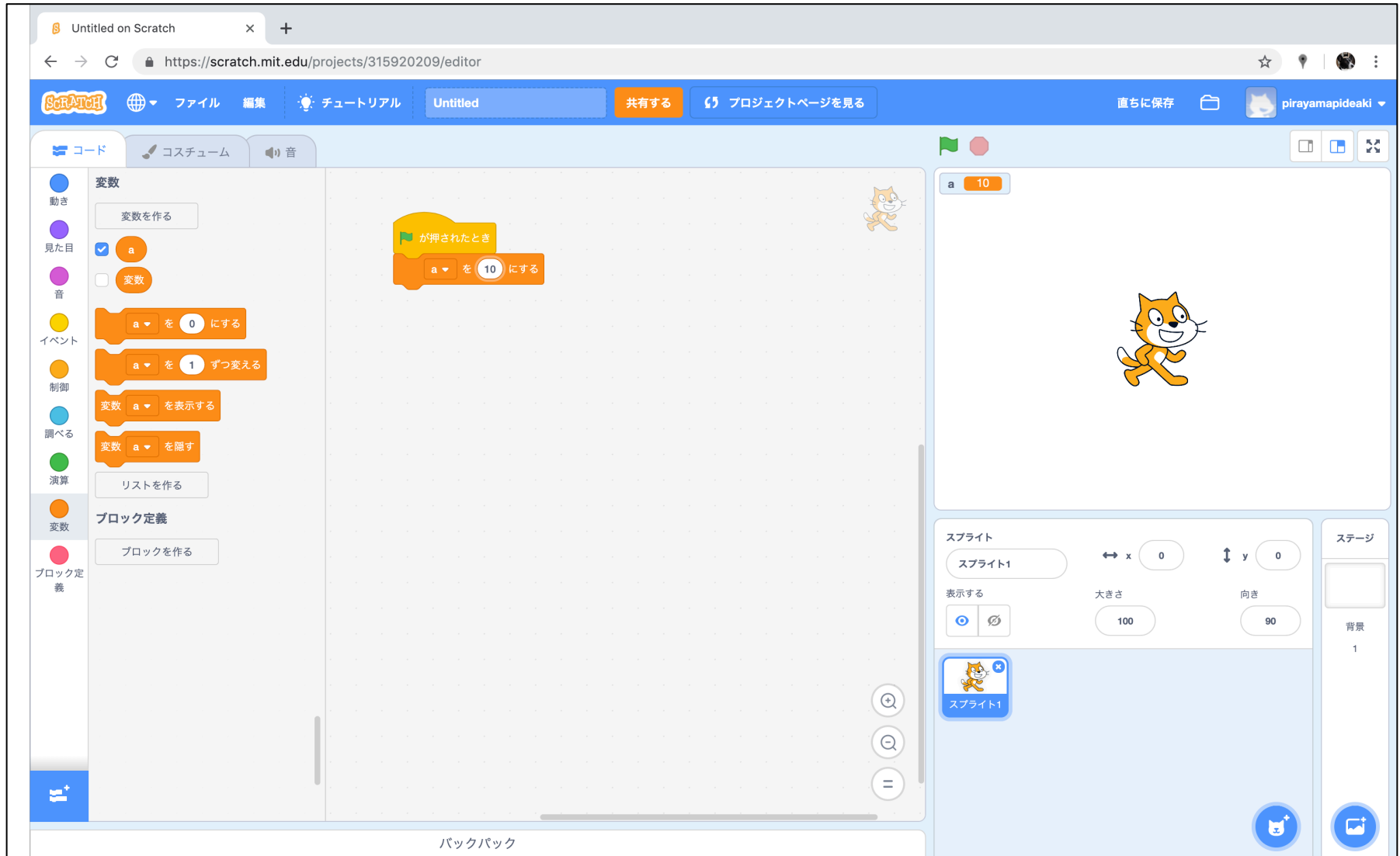
aは、変数です。

これは、「変数aと10は等しい」という意味ではなく、
「変数aに10を代入する(変数aを10にする)」という意味です。

- ・SCRATCHの変数の宣言は、こんな感じ。



・SCRATCHの代入は、こんな感じ。



- ・プログラミングの勉強では次に変数の演算について学びます。
以下のような文です。

```
y = 10 * x + 5;
```

これは、「10と変数xをかけ算した結果に5を足し、その結果をyに代入する」

という意味です。

プログラミングにおける四則演算の記号は、
以下のようになります。

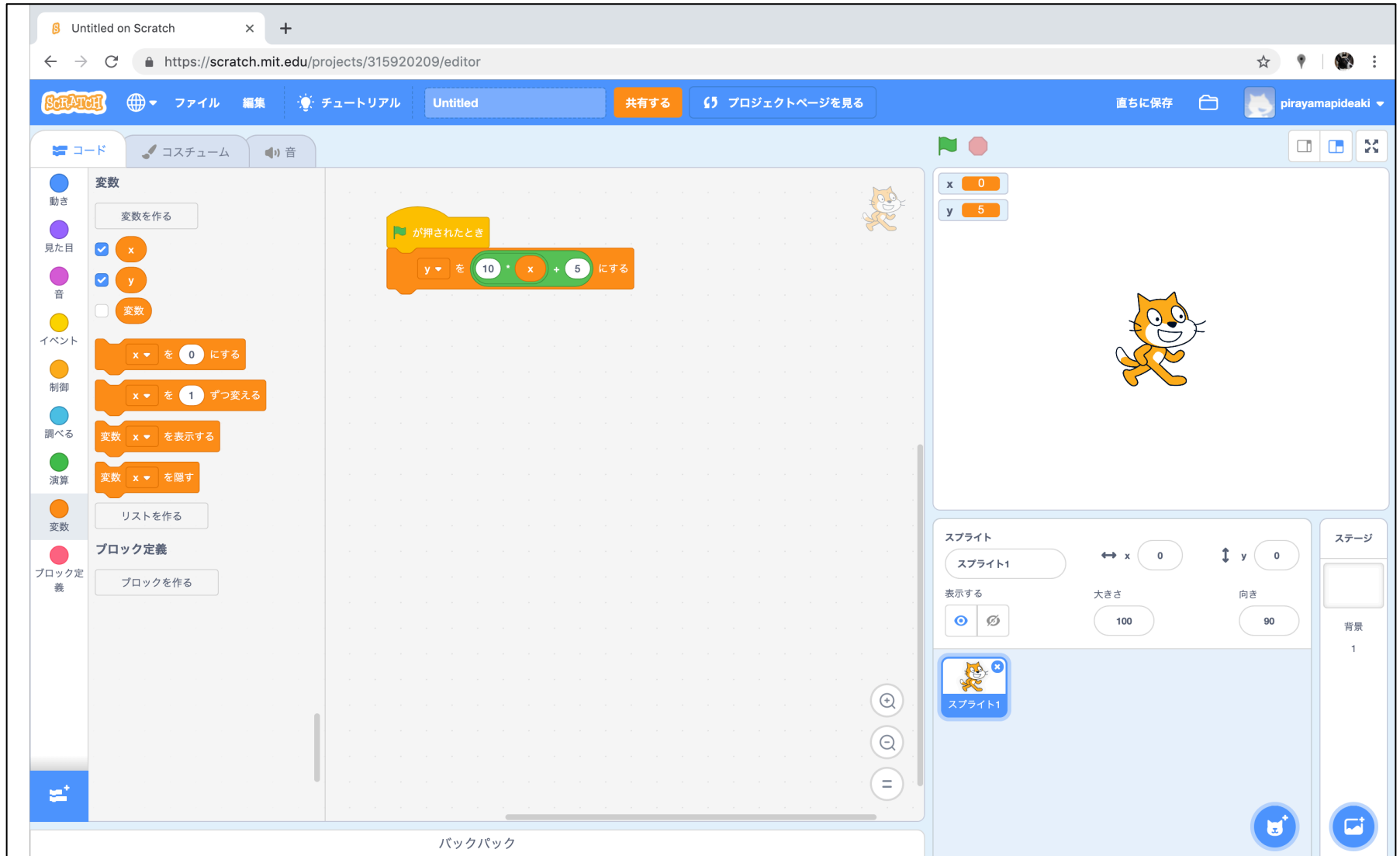
「+」は、「+」

「-」は、「-」

「×」は、「*」

「÷」は、「/」

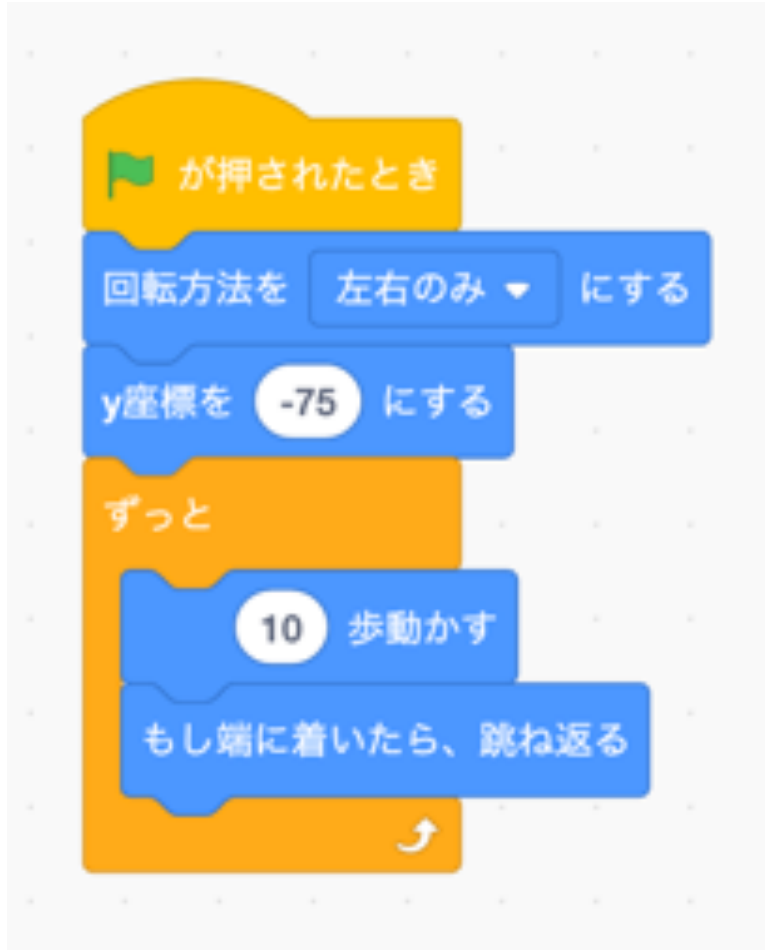
・SCRATCHの四則演算は、こんな感じ。



- ・コンピュータを英語で書くとcomputerです。
computerとは、computeする機械です。
computeとは、「計算する」という意味です。
- ・コンピュータは元々、軍事目的で開発されました。
ミサイルを正確に目的地に落とすための計算を行うとか。
- ・その後、コンピュータは商用目的で利用されるようになりました。
売上の集計とかの計算のためです。
人間より、はるかに速く、はるかに正確に、
単純な売上の集計とかの計算をひたすら繰り返します。
- ・コンピュータが得意なのが、繰り返しです。
第1回で、「ずっと」という繰り返しを、
第2回で、「<>まで」という繰り返しを学びました。
これは、普通のプログラムでは、while文というものでした。
ちょっと、第1回、第2回を振り返ってみましょう。

第1回より

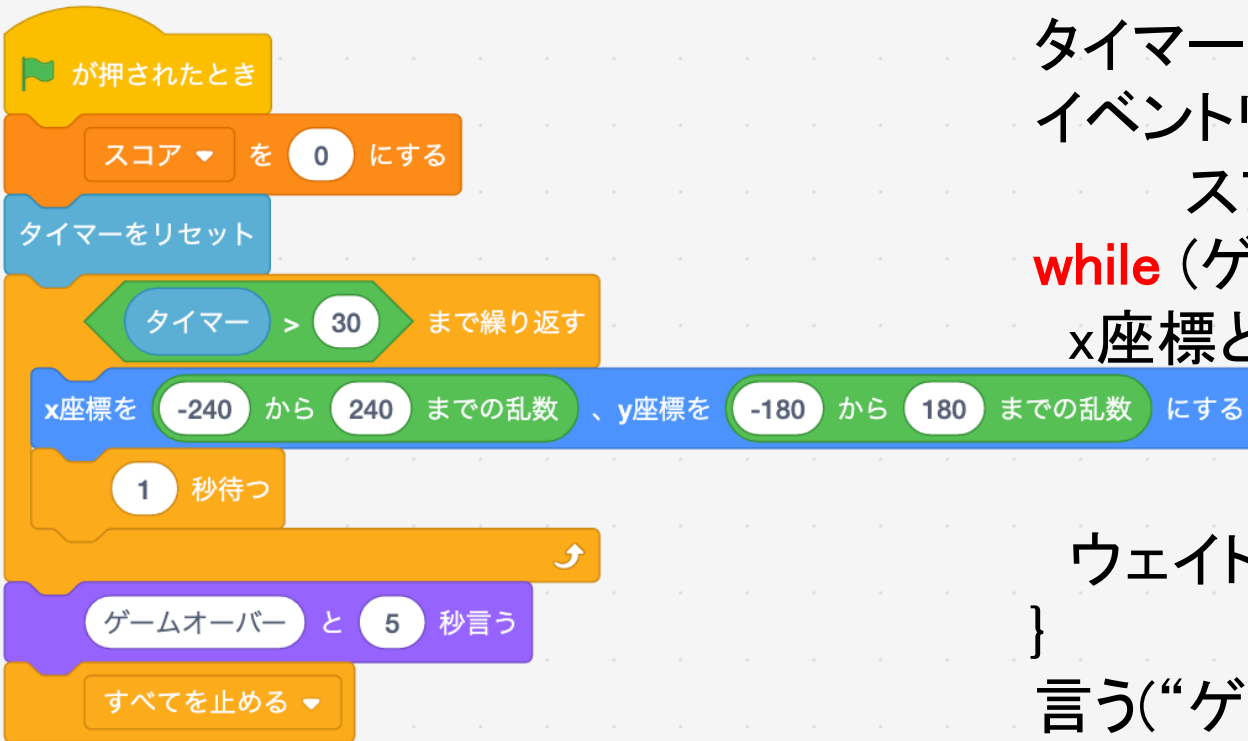
- ・ちょっと考えてみましょう。
今まで作ってきたブロックプログラムを、
普通のプログラムで書くとこんな感じになります。



```
main()
{
    回転方向を設定する(左右のみ);
    y座標を設定する(-75);
    while (1) {
        動かす(10歩);
        if (端に着いた) 跳ね返る();
    }
}
```

第2回より

- ・ちょっと考えてみましょう。
今まで作ってきたブロックプログラムを、
普通のプログラムで書くとこんな感じになります。



```
main()
{
    スコア = 0;
    タイマーリセット();
    イベントリスナー設定(イベント処理
                          スプライトがクリックされた);
    while (ゲットタイマー() > 30) {
        x座標とy座標を設定する(
            乱数(-240, 240)
            乱数(-180, 180)
        );
        ウェイト(1);
    }
    言う("ゲームオーバー", 5);
    exit();
}
```

- 実は、普通のプログラムのwhile文と、
SCRATCHの「」は、少し意味が異なります。



普通のプログラムのwhile文は、
「条件が成立している間は繰り返す」という意味なのに対し、
SCRATCHの「」は、



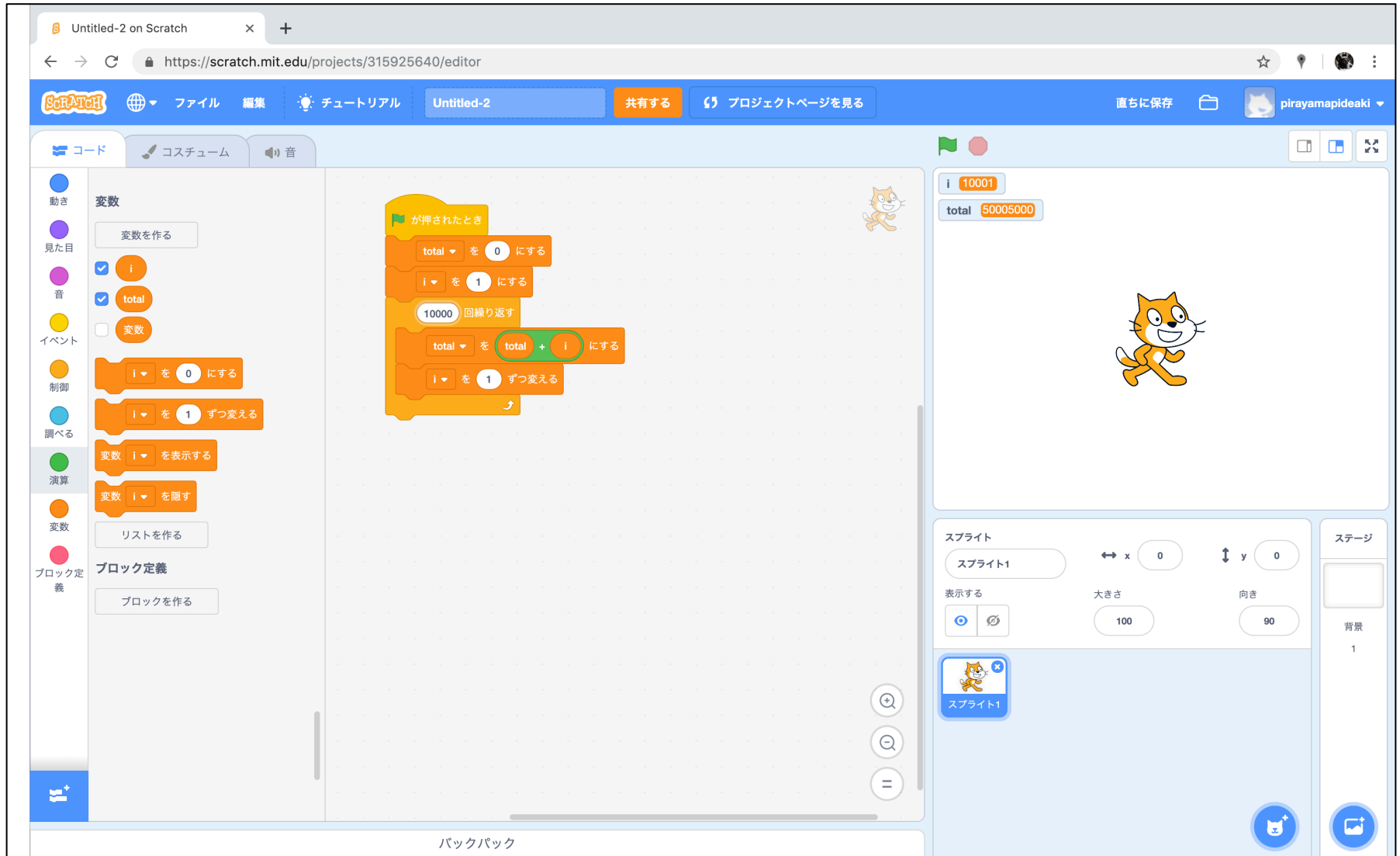
「条件が成立するようになるまで繰り返す」という意味なのです。

- 条件が逆ですね。
コンピュータというものは、人間のように曖昧なものを
上手く解釈するようなことはできません。
- コンピュータ、プログラミングの基本には、
ものごとを正確に表現するための論理演算というものが
使われています。
プログラムでコンピュータを動かすには、
正確に表現する、正確に指示することが必要です。

- ・少し話が理屈っぽくなったかもしれませんが。
さて、普通のプログラミング言語には、
while文の他に、for文という繰り返し文があります。
以下は、1から10000までの整数の合計を計算するプログラムです。

```
total = 0;
for (i = 1; i <= 10000; i++) {
    total = total + i;
}
```

- ・SCRATCHで、1から10000までの整数の合計を計算するプログラムは、こんな感じ。



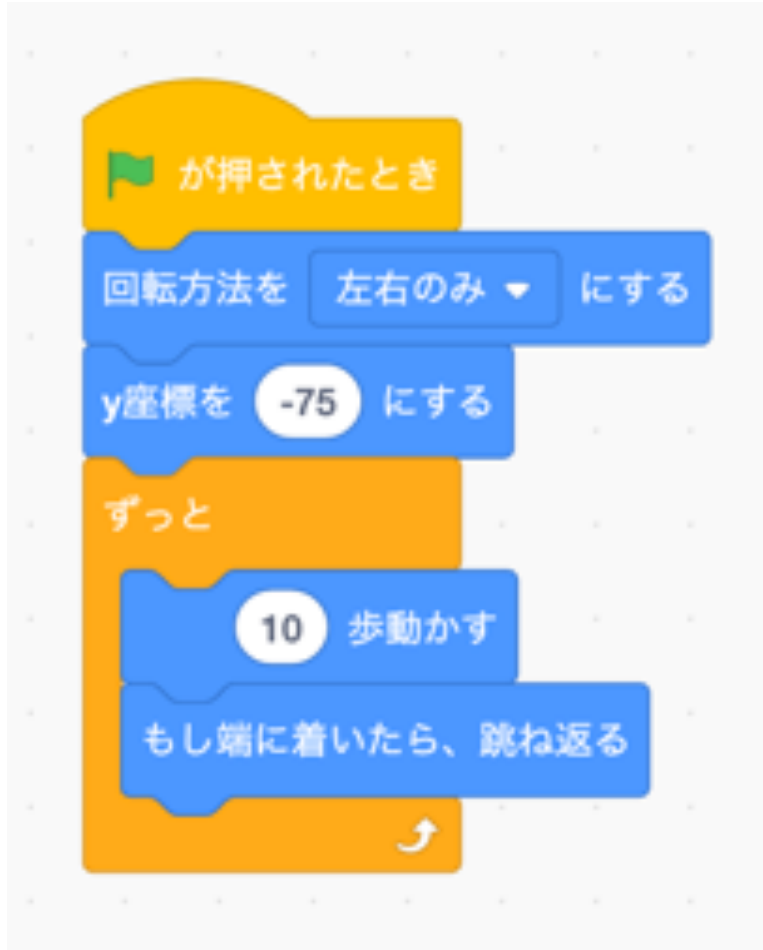
- ・もう一言付け加えると、普通のプログラムは、
1から始めるのではなく、0から始めるのが一般的です。
その方が元々のコンピュータの原理に向いていたのと、
その後も慣習的に、それが定着したことによります。
すなわち、普通のプログラムでは、
以下のようなfor文が書くのが一般的です。

```
total = 0;  
for (i = 0; i < 10000; i++) {  
    total = total + i;  
}
```

勿論、上記のプログラムは、
1から10000までの整数の合計ではなく、
0から9999までの整数の合計を計算することになりますが。

- ・あと、第1回で、第2回で、「もし」という処理を学びました。
これは、「条件判断」というもので、
普通のプログラムでは、if文というものです。
- ・「変数の宣言」、「代入」、「繰り返し」、「条件判断」は、
コンピュータ、プログラミングの基本要素です。

- ・ちょっと考えてみましょう。
今まで作ってきたブロックプログラムを、
普通のプログラムで書くとこんな感じになります。



```
main()
{
    回転方向を設定する(左右のみ);
    y座標を設定する(-75);
    while (1) {
        動かす(10歩);
        if (端に着いた) 跳ね返る();
    }
}
```

・続き。

第2回より

このスプライトが押されたとき

もし タイマー < 30 なら

Ya ▼ の音を鳴らす

スコア ▼ を 1 ずつ変える

イベント処理()

```
{  
  if (ゲットタイマー() < 30) {  
    音を鳴らす("Ya");  
    スコア = スコア + 1;  
  }  
}
```

- 少し難しくなりますが、
自分でシューティングゲームを作ってみましょう。
- 登場人物は、
フットボーラー (Football)
コウモリ (Bat)
ボール (Ball)
の3人です。
- ステージは、
Blue Sky
にします。
- 空を飛ぶコウモリに対し、
フットボーラーがボールを投げ、
コウモリの命がゼロになったらゲームクリアです。
フットボーラが投げるボールがなくなったらゲームオーバーです。

- ・基本的な動作は以下ようになります。
- ・コウモリは自分で空を飛び、
右端まで行ったら向きを変え、
左端まで行ったら再び向きを変え、
飛び続けます。
- ・フットボーラーは自分では動かず、
左右の矢印キーを押すことで、
プレイヤーが動かします。
- ・ボールは、スペースキーを押した時に発射され、
コウモリに当たったら、コウモリの命(最初は5個)を減らします。
コウモリの命が無くなったらゲームクリアです。
ボールが画面の端まで行ったら、
ボールの数(最初は10個)を減らします。
ボールが無くなったら、ゲームオーバーです。

- ・登場人物毎に動作を考えていきましょう。
- ・まずは、フットボーラーです。
フットボーラーの動作は以下です。
- ・旗が押されたら、
フットボーラーを適度なサイズにします。
また、x座標、y座標を適度な位置にします。
持っているボールの数を10個にします。
- ・ボールの数が0かどうか調べ続けます。
もし、ボールの数が0になったら、
ゲームオーバーと5秒言って、
全てを止めます。
- ・左向き矢印を押されたら、左に-10歩動きます。
- ・右向き矢印を押された場合も、同様の動きます。

- ・次に、コウモリです。
コウモリの動作は以下です。
- ・旗が押されたら、
コウモリを適度なサイズにします。
また、x座標、y座標を適度な位置にします。
端にぶつかってもひっくり返らないようにします。
持っている命の数を5個にします。
- ・コウモリは、ずっと10歩ずつ動き続けます。
端に付いたら逆向きに飛ぶようにします。
もし、命の数が0になったら、
ゲームオーバーと5秒言って、
全てを止めます。

- ・最後に、ボールです。
ボールの動作は以下です。
- ・旗が押されたら、
ボールを適度なサイズにします。
また、最初は見えないようにしておきます。
- ・スペースキーが押されたらボールを発射します。
ただし、ボールの数が0だったら発射しません。
発射する時は、ボールをフットボーラーの位置にして、
見えるようにします。
ボールは、少しずつ上に移動します。
- ・ボールがコウモリにぶつかったら、
コウモリの命の数を1減らして、
ボールを見えないようにします。
- ・ボールが端まで行ったら、
ボールの数を1減らして、
ボールを見えないようにして、
ボールの動作を止めます。

- ・さあ、やってみましょう。

- ・できましたか？
完全にできなくても構いません。
自分で考えてやってみることが大切です。
- ・教科書に載っているプログラムを見たり、
真似したりしながら学んだら、
次は、自分で考えてやってみます。
それを繰り返していくことで、
プログラミングができるようになります。
- ・シューティングゲームのプログラムを、
次ページ以降に示します。

・フットボーラのプログラム

The screenshot shows a Scratch project titled "第3章パート1 on Scratch" (Chapter 3 Part 1 on Scratch). The code is written in Japanese and includes the following logic:

- When the green flag is clicked:**
 - Set the size of the ball to 40%.
 - Set the x-coordinate to 0 and the y-coordinate to -120.
 - Set the ball's count to 10.
- Forever loop:**
 - Set the ball's count to 0.
 - Increment the ball's count by 1.
 - Display the ball's count.
 - Hide the ball's count.
- When the left arrow key is pressed:**
 - Move the ball 10 steps to the left.
- When the right arrow key is pressed:**
 - Move the ball 10 steps to the right.
- When the ball's count reaches 0:**
 - Say "ゲームオーバー" (Game Over) for 5 seconds.
 - Stop everything.

The right side of the image shows the game's visual representation. A football player is on a green field with a blue sky background. A bat is flying in the sky. The ball is a yellow circle. The player's position is at x=0, y=-120. The ball's position is at x=0, y=-120. The ball's size is 40. The ball's count is 10. The game is over when the ball's count reaches 0.

・コウモリのプログラム

第3章パート1 on Scratch

scratch.mit.edu/projects/336642265/editor

コード コスチューム 音

変数

動き

見た目

音

イベント

制御

調べる

演算

変数

ブロック定義

ブロックを作る

リストを作る

変数を作る

ボール

変数

命

ボール を 0 にする

ボール を 1 ずつ変える

変数 ボール を表示する

変数 ボール を隠す

旗が押されたとき

大きさを 40 %にする

x座標を 0 、y座標を 155 にする

回転方法を 左右のみ にする

命 を 5 にする

ずっと

10 歩動かす

もし端に着いたら、跳ね返る

もし 命 = 0 なら

ゲームクリア と 5 秒言う

すべてを止める

旗が押されたら、
コウモリの大きさと位置をセットします。
コウモリの回転方向を左右のみにします。
コウモリの命の数を5にします。

コウモリは、ずっと、10歩ずつ動き続けます。
端についたら跳ね返ります。

もし、命の数が0になったら、
ゲームクリアと5秒言って、
全てを止めます。

ボール 10

命 5

スプライト

Bat

表示する

大きさ 40

向き -90

Football

Bat

Ball

ステージ

背景 2

バックバック

・ボールのプログラム

第3章パート1 on Scratch

scratch.mit.edu/projects/336642265/editor

第3章パート1

共有する

プロジェクトページを見る

直ちに保存

hirayama968

コード

変数

動き

見た目

音

イベント

制御

調べる

演算

変数

ブロック定義

ブロックを作る

変数を作る

ボール

変数

命

ボール を 0 にする

ボール を 1 ずつ変える

変数 ボール を表示する

変数 ボール を隠す

リストを作る

ブロック定義

ブロックを作る

旗が押されたとき

大きさを 40 %にする

隠す

スペース キーが押されたとき

もし ボール > 0 なら

Football へ行く

表示する

ずっと

y座標を 10 ずつ変える

もし Bat に触れた なら

命 を -1 ずつ変える

隠す

もし 端 に触れた なら

ボール を -1 ずつ変える

隠す

このスクリプトを止める

ボール 10

命 5

旗が押されたら、
ボールの大きさをセットします。
ボールは見えなくしておきます。

スペースキーが押されたら、以下の処理を行います。
もし、ボールの数が0より大きければ、以下の処理を行います。
フットボーラーの位置にボールを表示します。
ずっと、以下の処理を繰り返します。
y座標を10ずつ変えます。
もし、コウモリに触れたら、
命の数を1減らし、
ボールを隠します。
もし、端に触れたら、
ボールの数を1減らし、
ボールを隠します。
そして、このスクリプトを止めます。

スプライト

Ball

表示する

大きさ 40

向き 90

Football

Bat

Ball

ステージ

背景 2

バックバック

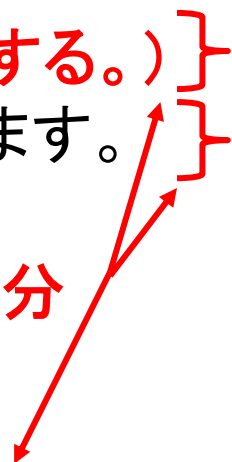
- ・最初からお手本のプログラムを見るのと、最初に自分で作ってみようとしてからお手本のプログラムを見るのでは、ちょっと違いますか？
- ・もう少し、プログラムを直してみます。
このままだと、フットボーラーを動かさずに、タイミングだけ見てボールを投げるのが、最適な攻略法です。
でもそれでは面白くありません。
フットボーラーを全く動かさない場合は、コウモリの動きを速くします。
フットボーラを動かしたら、コウモリの動きを元の速さに戻します。
プログラムの動作は、以下のように変わります。
(変数「速度」を作り、コウモリの動きを、
10歩から「速度」歩にします。)

- ・登場人物毎に動作を考えていきましょう。
- ・まずは、フットボーラーです。
フットボーラーの動作は以下です。
- ・旗が押されたら、
フットボーラーを適度なサイズにします。
また、x座標、y座標を適度な位置にします。
持っているボールの数を10個にします。
- ・ボールの数が0かどうか調べ続けます。
もし、ボールの数が0になったら、
ゲームオーバーと5秒言って、
全てを止めます。
- ・左向き矢印を押されたら、左に-10歩動きます。
- ・右向き矢印を押された場合も、同様の動きます。
- ・左でも、右でも、フットボーラーが動いた時には、
コウモリの速度を最初の速度に戻します。
(変数「速度」を10にする。)

} 変更部分

- ・次に、コウモリです。
コウモリの動作は以下です。
- ・旗が押されたら、
コウモリを適度なサイズにします。
また、x座標、y座標を適度な位置にします。
端にぶつかってもひっくり返らないようにします。
持っている命の数を5個にします。
コウモリの最初の速度を10にします。(変数「速度」を10にする。) }
- ・コウモリは、ずっと**変数「速度」で示された速度で動き続けます。** }
端に付いたら逆向きに飛ぶようにします。
もし、命の数が0になったら、
ゲームオーバーと5秒言って、
全てを止めます。
- ・もし、**スペースキーを押された時は、** }
コウモリの速度を1速くします。(変数「速度」に1加える。) }

変更部分



- ・最後に、ボールです。
ボールの動作は以下です。
- ・旗が押されたら、
ボールを適度なサイズにします。
また、最初は見えないようにしておきます。
- ・スペースキーが押されたらボールを発射します。
ただし、ボールの数が0だったら発射しません。
発射する時は、ボールをフットボーラーの位置にして、
見えるようにします。
ボールは、少しずつ上に移動します。
- ・ボールがコウモリにぶつかったら、
コウモリの命の数を1減らして、
ボールを見えないようにします。
- ・ボールが端まで行ったら、
ボールの数を1減らして、
ボールを見えないようにして、
ボールの動作を止めます。

・フットボーラのプログラム

第3章パート2 on Scratch

scratch.mit.edu/projects/336642265/editor

hirayama968

コード コスチューム 音

変数

動き

見た目

音

イベント

制御

調べる

演算

変数

ブロック定義

ブロックを作る

変数を作る

ボール

速度

変数

命

ボール を 0 にする

ボール を 1 ずつ変える

変数 ボール を表示する

変数 ボール を隠す

リストを作る

ブロックを作る

旗が押されたとき

大きさを 40 %にする

x座標を 0、y座標を -120 にする

ボール を 10 にする

ずっと

もし ボール = 0 なら

ゲームオーバー と 5 秒言う

すべてを止める

左向き矢印 キーが押されたとき

-10 歩動かす

速度 を 10 にする

右向き矢印 キーが押されたとき

10 歩動かす

速度 を 10 にする

旗が押されたら、フットボーラーの大きさと位置、それと最初に持っているボールの数をセットします。

ボールの数が0かどうか調べ続けます。もし、ボールの数が0になったら、ゲームオーバーと5秒言って、全てを止めます。

左向き矢印を押されたら、左に10歩動かします。フットボーラーを動かしたので、コウモリの速度を最初の速度に戻します。(変数「速度」を10にする)

右向き矢印を押されたら、右に10歩動かします。フットボーラーを動かしたので、コウモリの速度を最初の速度に戻します。(変数「速度」を10にする)

ボール 10

命 5

スプライト

表示する

Football

Bat

Ball

ステージ

背景 2

バックバック

・コウモリのプログラム

第3章パート2 on Scratch

scratch.mit.edu/projects/336646024/editor

コード

動き

10 歩動かす

15 度回す

15 度回す

どこかの場所へ行く

x座標を -39、y座標を 155 にする

1秒でどこかの場所へ行く

1秒でx座標を -39に、y座標を 155に

90度に向ける

マウスのポインターへ向ける

x座標を 10 ずつ変える

x座標を -39 にする

y座標を 10 ずつ変える

y座標を 155 にする

が押されたとき

大きさを 40 %にする

x座標を 0、y座標を 155 にする

回転方法を 左右のみ にする

命 を 5 にする

速度 を 10 にする

ずっと

速度 歩動かす

もし端に着いたら、跳ね返る

もしスペースキーが押されたなら

速度 を 1 ずつ変える

もし命 = 0 なら

ゲームクリア と 5秒言う

すべてを止める

ボール 10

命 5

旗が押されたら、コウモリの大きさと位置をセットします。コウモリの回転方向を左右のみにします。コウモリの命の数を5にします。フットボーラーの最初の速度を10にします。(変数「速度」を10にします。)

コウモリは、ずっと、変数「速度」で示された歩数ずつ動き続けます。端についたら跳ね返ります。

もし、スペースキーを押されたら、変数「速度」に1加えます。

もし、命の数が0になったら、ゲームクリアと5秒言って、全てを止めます。

スプライト

Bat

Football

Ball

ステージ

背景 2

バックバック

・ボールのプログラム

第3章パート2 on Scratch

scratch.mit.edu/projects/336642265/editor

コード コスチューム 音

変数

変数を作る

ボール

速度

変数

命

ボール を 0 にする

ボール を 1 ずつ変える

変数 ボール を表示する

変数 ボール を隠す

リストを作る

ブロック定義

ブロックを作る

バックバック

旗が押されたとき

大きさを 40 %にする

隠す

スペース キーが押されたとき

もし ボール > 0 なら

Football へ行く

表示する

ずっと

y座標を 10 ずつ変える

もし Bat に触れた なら

命 を -1 ずつ変える

隠す

もし 端 に触れた なら

ボール を -1 ずつ変える

隠す

このスクリプトを止める

旗が押されたら、
ボールの大きさをセットします。
ボールは見えなくしておきます。

スペースキーが押されたら、以下の処理を行います。
もし、ボールの数が0より大きければ、以下の処理を行います。
フットボーラーの位置にボールを表示します。
ずっと、以下の処理を繰り返します。
y座標を10ずつ変えます。
もし、コウモリに触れたら、
命の数を1減らし、
ボールを隠します。
もし、端に触れたら、
ボールの数を1減らし、
ボールを隠します。
そして、このスクリプトを止めます。

ボール 10
命 5

スプライト

Ball

表示する

大きさ 40

向き 90

Football

Bat

Ball

ステージ

背景 2

- ・フットボーラー、コウモリ、ボール、
それぞれ個別にプログラミングを行いました。
全てをいっしょくたにしてしまうのではなく、
スプライト毎に動作を分けて考えました。
これをオブジェクト指向と言います。
必要なオブジェクト(スプライト)をきちんと定義し、
その定義したオブジェクト毎に動作を決めていくのです。
なんでもかんでもいっしょくたにしたら、
整理できません、まとまりません、
ぐちゃぐちゃになってしまいます。
規模が大きいほどそうなります。
- ・このオブジェクト指向の考え方は、
設計、プログラミングだけでなく、
あらゆるところで役に立ちます。

・最終回は少し理屈っぽかったですね。
また、シューティングゲームは、
少し難しかったですね。
是非、家で、お手本のプログラムを動かして、
やっていることを理解してみてください。
人が書いたプログラムを読むのも、
プログラミングの重要な勉強方法です。
また、自分でどのようなものを作るか考え、
それを実際にプログラミングしてみてください。
プログラミングは、そうやって身につけていきます。
設計は、規模の大きなプログラムを作るようになると、
その必要性が見えてきます。