

# micro:bitによるプログラミング (3)

担当：稲川孝司（帝塚山学院大学）

## 実習内容

- NeoPixelによるフルカラーLEDの制御

## 引用・参考文献

高橋参吉、喜家村奨、稲川孝司：micro:bitで学ぶプログラミング  
ブロック型からJavaScriptそしてPythonへ、コロナ社（2019.9）

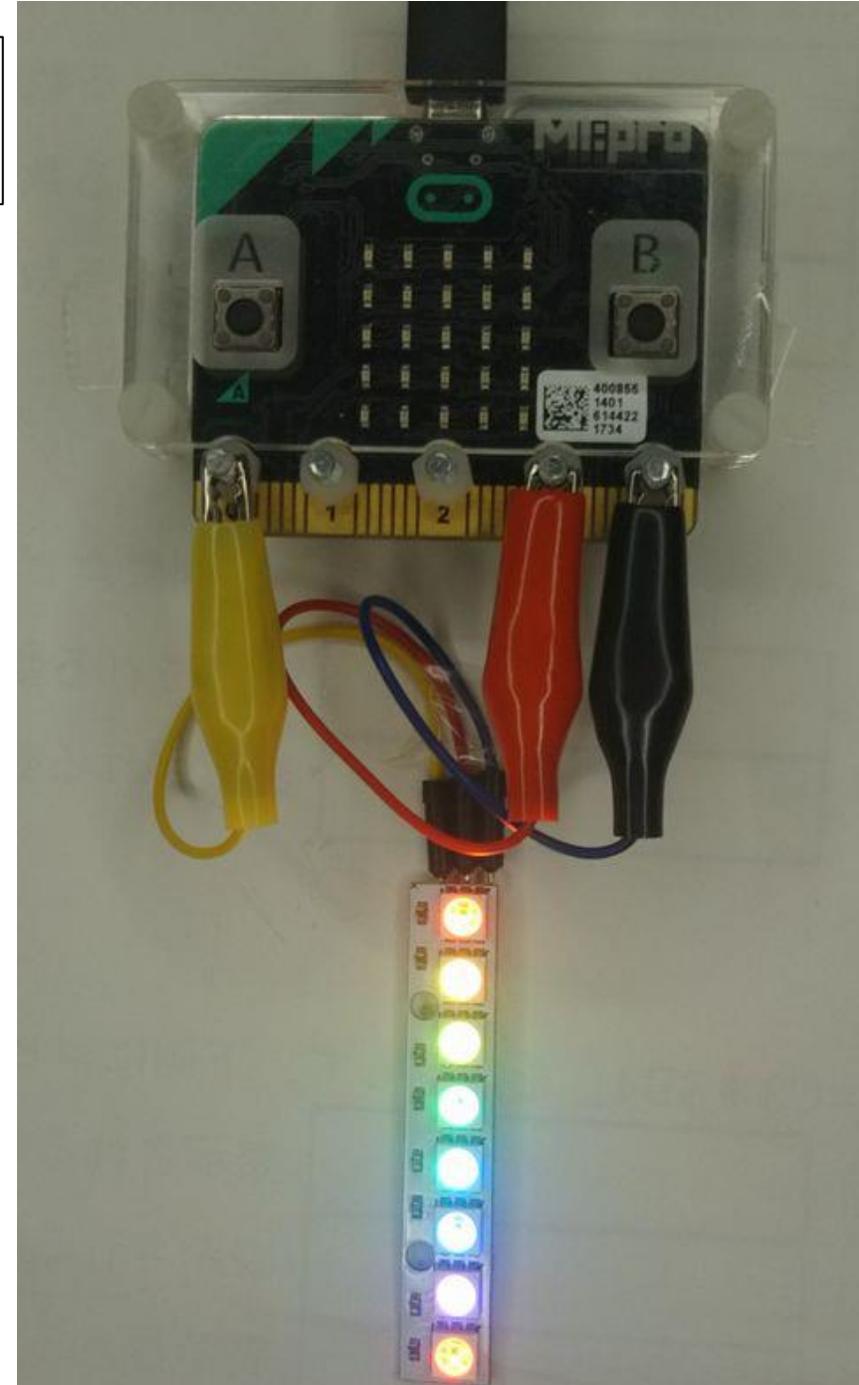
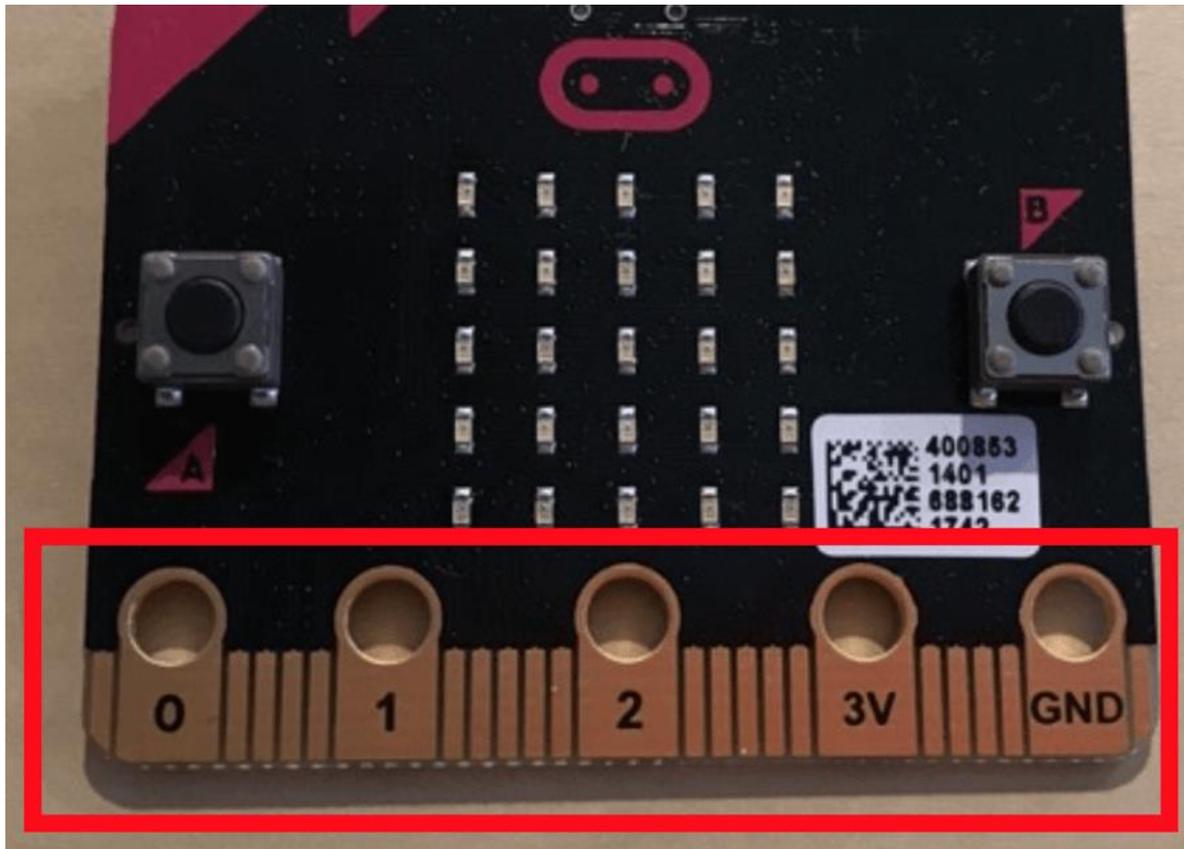
# NeoPixelとは

- NeoPixelは、1つのセル毎にRGBの3つのLEDとその制御回路が入っており、シリアル通信でRGBの値を送ることで、簡単に多くのLEDをフルカラーで光らせることができるLEDの集合体です。RGBそれぞれを256段階に調整でき、 $256*256*256=16777216$ 色で光ります。
- 今回は、8個のLEDがStrip型に接続された製品を使って、様々な色で光らせる実習を行います。
- micro:bitとNeopixelがなくても、ブラウザのみで実習は可能です。その場合、最後のmicro:bitに書き込む作業は不要です。

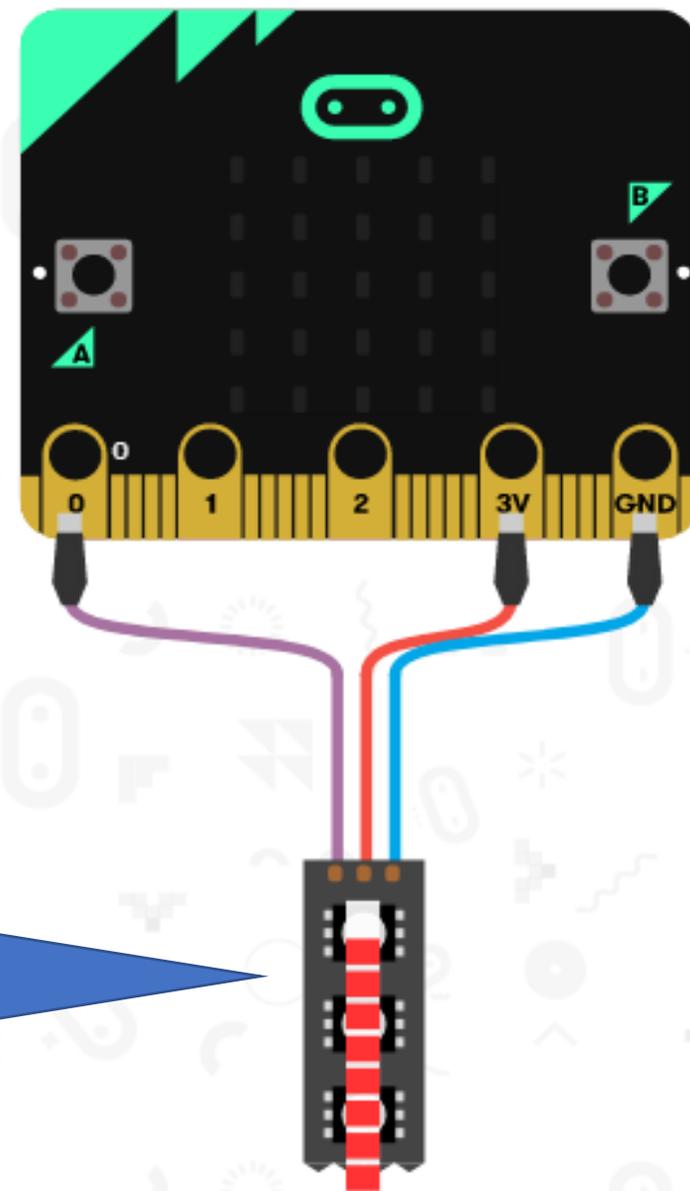
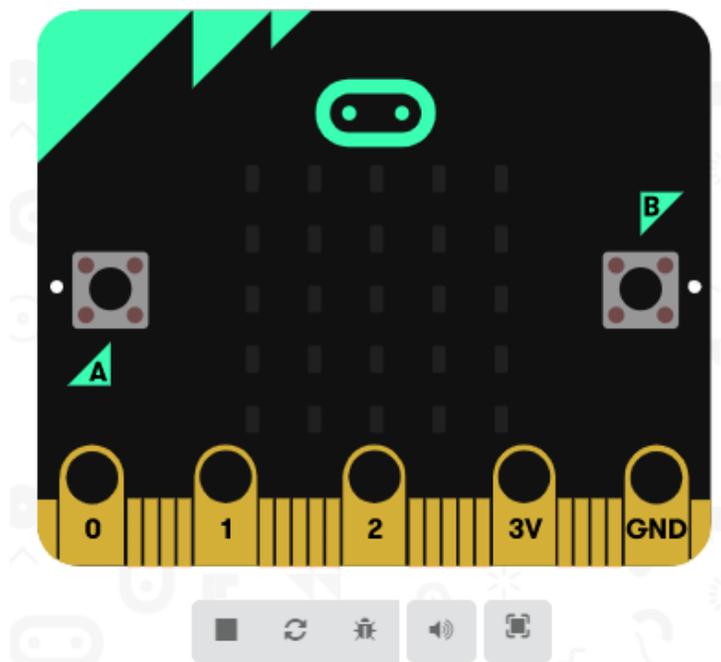


# Micro:bitとNeoPixelの接続

Micro:bitとNeoPixelを右図のように接続します。  
黄色はP0, 赤は3V, 黒はGND



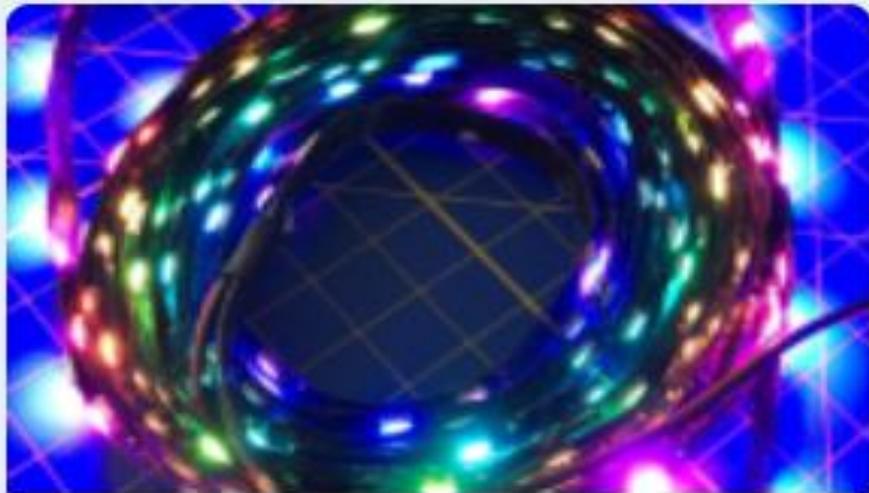
# シミュレータについて



NeoPixelライブラリを追加し、プログラムを作成すると、シミュレータ画面に自動的にNeoPixelが表示されます。

# 【準備】NeoPixelライブラリーの入手

- 1) ブラウザで <https://makecode.microbit.org> サイトに行く。
- 2) 新しいプロジェクト⇒名前を「rei3-1」と入力し、作成ボタンをクリックする。
- 3) ツールボックスの一番下の「高度なブロック」⇒「+拡張機能」  
⇒「neopixel」の**写真をクリック**する。
- 4) ツールボックス内に、水色のNeoPixelのブロック群が追加される。



neopixel



## 【例題3-1】 NeoPixelを光らせてみよう。(ファイル名:rei3-1)

### <手順>

- 1) 新しいプロジェクト⇒名前を「rei3-1」と入力し、作成ボタンをクリックする。
- 2) 「最初だけ」ブロックの中に「変数をNeoPixelモードにする」を追加する。
- 3) NeoPixelを接続している端子はP0なので、P0であることを**確認**する。
- 4) NeoPixelの数は8個なので、ブロック内の24を8に**変更**する。
- 5) 「Neopixel」ブロックから「赤色に点灯する」ブロックを「最初だけ」に追加する
- 6) **作成したプログラムをmicro:bitに書き込み**、赤色で点灯することを確認する。

最初だけ

変数 strip を 端子 P0 に接続しているLED 8 個のNeoPixel (モード RGB (GRB順)) にする

strip を 赤 色に点灯する

24を8に変更

ここをクリックして、好きな色に変えることができる

## 【例題3-2】 LEDを好きな色で100ミリ秒ごとに点滅させてみよう。 (ファイル名:rei3-2)

### <手順>

- 1) ファイル名を「rei3-2」と入力する。
- 2) 「最初だけ」ブロックの中に「変数をNeoPixelモードにする」ブロックを追加する
- 3) NeoPixelを接続している端子はP0なので、P0であることを確認する。
- 4) NeoPixelの数は8個なので、ブロック内の24を8に**変更**する。  
-----
- 5) 「ずっと」ブロックの中に「赤色に点灯する」ブロックを追加する。
- 6) 基本から「一時停止（ミリ秒）100」ブロックを追加する
- 7) 「赤色に点灯する」ブロックを追加し、消灯するために**色をblackに変更**する。
- 8) 再度、基本から「一時停止（ミリ秒）100」ブロックを追加する
- 9) **作成したプログラムをmicro:bitに書き込みます。**

## 【例題3-2】 (ファイル名:rei3-2)

最初だけ

変数 `strip` を (端子 `P0`) に接続しているLED (8) 個のNeoPixel (モード `RGB (GRB順)`) にする

ずっと

`strip` を `赤` 色に点灯する

一時停止 (ミリ秒) `100`

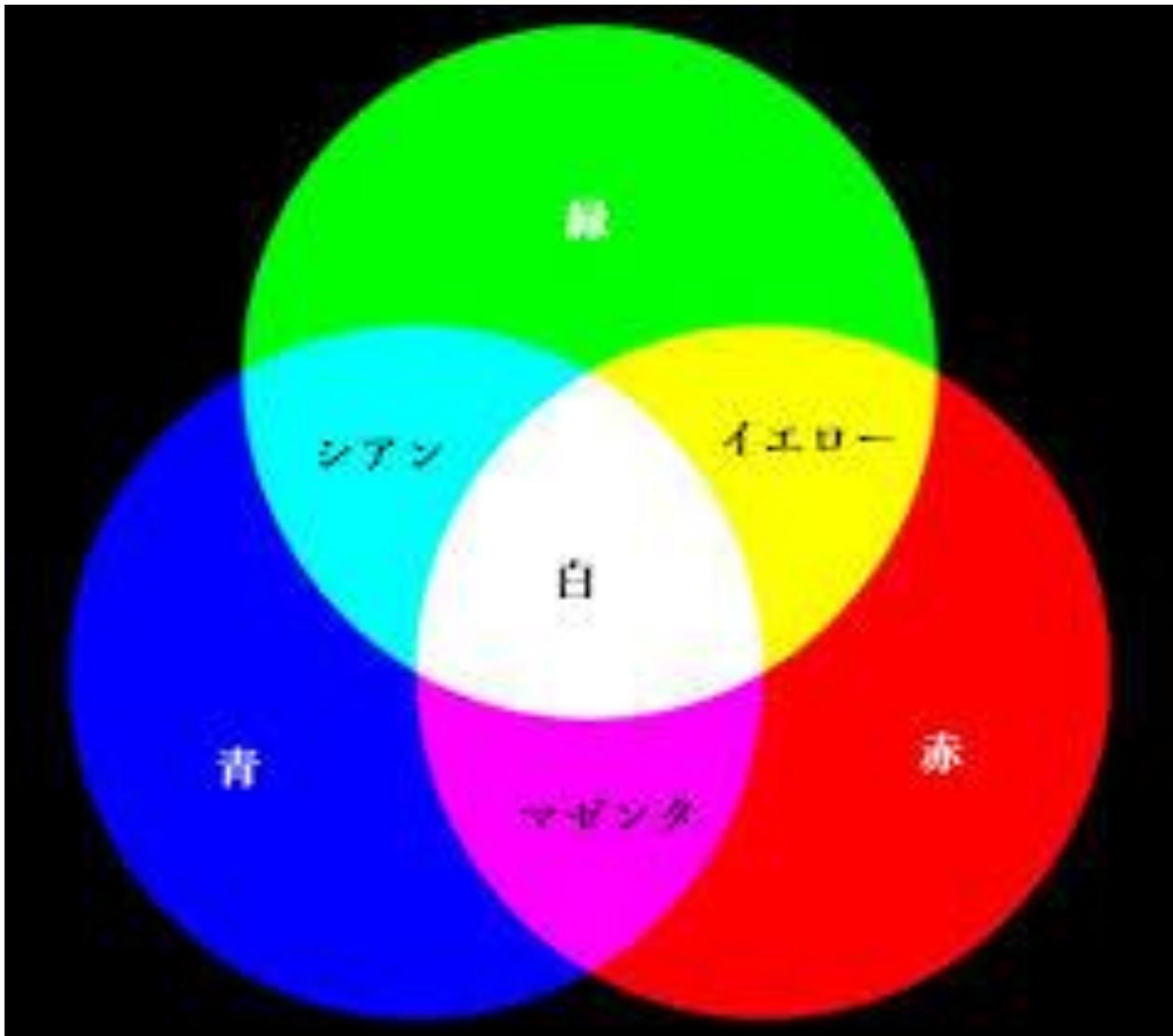
`strip` を `black` 色に点灯する

一時停止 (ミリ秒) `100`

Neopixelのその他ブロックからRGB(赤255、緑255、青255)ブロックを選ぶと、RGBそれぞれを256段階に変えて表示させることができる。

RGB (赤 255 緑 255 青 255 )

# 光の3原色 R(赤) G(緑) B(青)



赤 緑 青  
256色 × 256色 × 256色  
= 16777216色  
≒ 1677万色

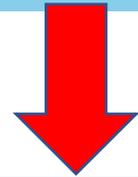
10進数 ⇔ 16進数  
0 ~ 255 ⇔ 00 ~ FF

|                           |                           |                              |                          |                         |
|---------------------------|---------------------------|------------------------------|--------------------------|-------------------------|
| floralwhite<br>#fffaf0    | blue<br>#0000ff           | palegreen<br>#98fb98         | darkorange<br>#ff8c00    | thistle<br>#d8bfd8      |
| linen<br>#faf0e6          | dodgerblue<br>#1e90ff     | lightgreen<br>#90ee90        | goldenrod<br>#daa520     | magenta<br>#ff00ff      |
| antiquewhite<br>#faebd7   | cornflowerblue<br>#6495ed | springgreen<br>#00ff7f       | peru<br>#cd853f          | fuchsia<br>#ff00ff      |
| papayawhip<br>#ffefd5     | deepskyblue<br>#00bfff    | mediumspringgreen<br>#00fa9a | darkgoldenrod<br>#b8860b | violet<br>#ee82ee       |
| blanchedalmond<br>#ffe4cd | lightskyblue<br>#87cefa   | lawngreen<br>#7cfc00         | chocolate<br>#d2691e     | plum<br>#dda0dd         |
| bisque<br>#ffe4c4         | skyblue<br>#87ceeb        | chartreuse<br>#7fff00        | sienna<br>#a0522d        | orchid<br>#da70d6       |
| moccasin<br>#ffe4b5       | lightblue<br>#add8e6      | greenyellow<br>#adff2f       | saddlebrown<br>#8b4513   | mediumorchid<br>#ba55d3 |
| navajowhite<br>#ffdead    | powderblue<br>#b0e0e6     | lime<br>#00ff00              | maroon<br>#800000        | darkorchid<br>#9932cc   |
| peachpuff<br>#ffdab9      | paleturquoise<br>#afeeee  | limegreen<br>#32cd32         | darkred<br>#8b0000       | darkviolet<br>#9400d3   |
| mistyrose<br>#ffe4e1      | lightcyan<br>#e0ffff      | yellowgreen<br>#9acd32       | brown<br>#a52a2a         | darkmagenta<br>#8b008b  |
| lavenderblush<br>#fff0f5  | cyan<br>#00ffff           | darkolivegreen<br>#556b2f    | firebrick<br>#b22222     | purple<br>#800080       |

# micro:bitでの16進数から10進数への自動変換】

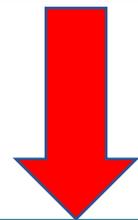
(例)

skyblue  
#87ceeb



RGB (赤 **0x87** 緑 **0xce** 青 **0xeb** )

16進数であることを示す0xを先頭につける



RGB (赤 **135** 緑 **206** 青 **235** )

自動的に10進数になる

# 【コラム：LEDの番号の割り当てについて】

NeoPixelは、順番にLEDを点灯させるために、それぞれに番号が割り当てられています。8個のバー型の場合は、右図のように、信号線に近いほうから順に0から7が割り当てられます。

個別に色を設定して点灯する方法は、

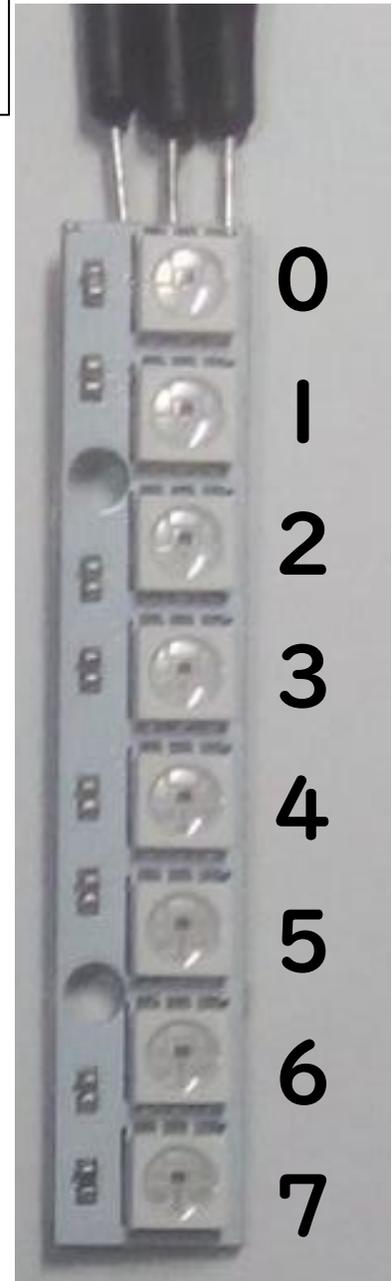
strip ▾ の 2 番目のLEDを 黄 ▾ 色に設定する

で番号を指定し、n番目のLEDを好きな色で点灯させることができます。

また、

strip ▾ に設定されている色をLED 1 個分ずらす

で、点灯しているLEDを下にずらすことができます。



## 【例題3-3】 8個のLEDを個別に好きな色に指定して、点灯させてみよう。 (ファイル名:rei3-3)

### <手順>

- 1) ファイル名を「rei3-3」と入力する。
- 2) 「最初だけ」ブロックの中に「変数をNeoPixelモードにする」ブロックを追加する。
- 3) 今回のNeoPixelの数は8個なので、ブロック内の24を8に変更する。  
-----
- 4) 「ずっと」ブロックの中に、NeoPixelの「その他」ブロックから「stripの0番目のLEDを赤色に設定する」ブロックを追加する。
- 5) 「設定」ブロックを右クリックして複製し、計8個の「設定」ブロックを作成。
- 6) 設定する番号を0から7の順に変更し、色は好きな色に変更する。
- 7) 「stripを設定した色で点灯する」ブロックを追加する。
- 8) 作成したプログラムをmicro:bitに書き込みます。

## 【例3-3】 (rei3-3)

最初だけ

変数 `strip` を 端子 `P0` に接続しているLED `8` 個のNeoPixel (モード `RGB (GRB順)` ) にする

ずっと

`strip` の `0` 番目のLEDを `赤` 色に設定する

`strip` の `1` 番目のLEDを `だいたい` 色に設定する

`strip` の `2` 番目のLEDを `黄` 色に設定する

`strip` の `3` 番目のLEDを `緑` 色に設定する

`strip` の `4` 番目のLEDを `青` 色に設定する

`strip` の `5` 番目のLEDを `あい` 色に設定する

`strip` の `6` 番目のLEDを `すみれ` 色に設定する

`strip` の `7` 番目のLEDを `紫` 色に設定する

`strip` を設定した色で点灯する

ブロックの1個目をコピーして、貼り付け、色を変える

**【例題3-4】 「1個分ずらす」ブロックを使って、点灯した緑のLEDを上から下へ移動させてみよう。 (ファイル名: rei3-4)**

<手順>

- 1) ファイル名を「rei3-4」と入力します。
- 2) 「最初だけ」中に「変数をNeoPixelモードにする」ブロックを追加する。
- 3) 今回のNeoPixelの数は8個なので、ブロック内の24を**8に変更**します。  
-----
- 4) 「ずっと」の中に、NeoPixelの「その他」ブロックから「0番目のLEDの色を設定する」ブロックを追加する。
- 5) ループから「くりかえし」ブロックを入れ、回数を8回に設定します。
- 6) 「くりかえし」の中に、NeoPixelから「指定した色で点灯する」ブロックを追加する。
- 7) さらに、基本から「一時停止(ミリ秒)100」ブロックを追加する。
- 8) そのブロックの下に、NeoPixelから「色をLED1個分ずらす」ブロックを追加する。
- 9) **作成したプログラムをmicro:bitに書き込みます。**

最初だけ

## 【例題3-4】 (rei3-4)

変数 `strip` を 端子 `P0` に接続しているLED (8) 個のNeoPixel (モード `RGB (GRB順)`) にする

ずっと

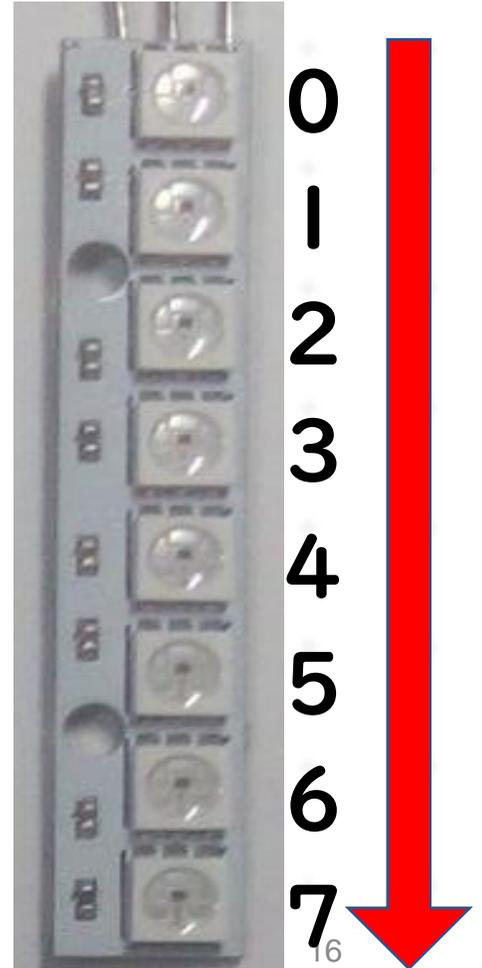
`strip` の 0 番目のLEDを 緑 色に設定する

くりかえし 8 回

`strip` を設定した色で点灯する

一時停止 (ミリ秒) 100

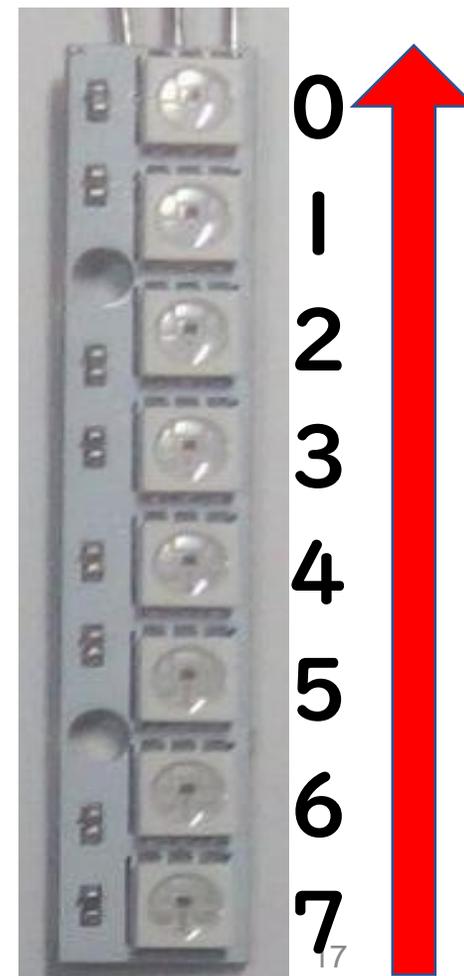
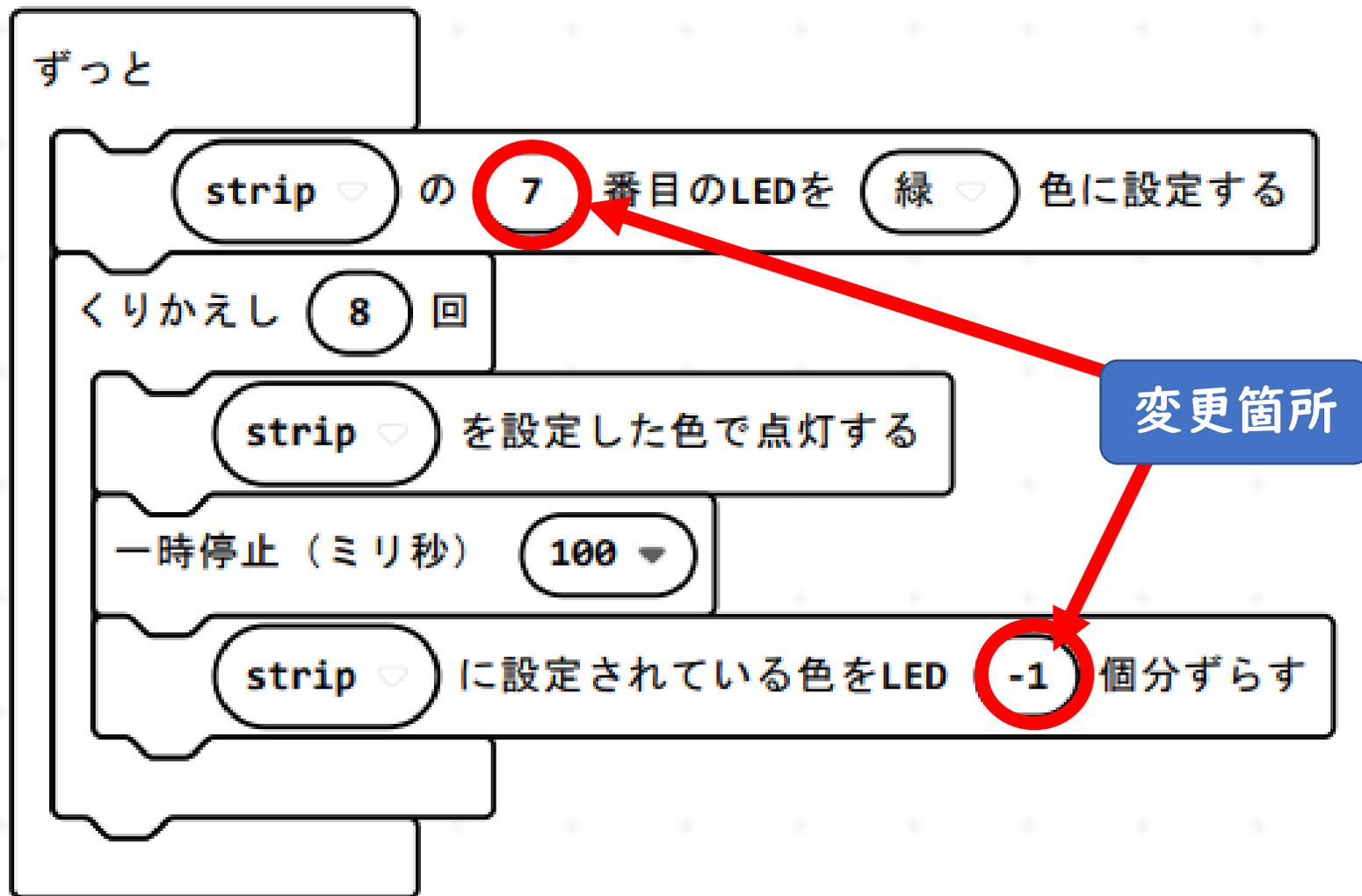
`strip` に設定されている色をLED 1 個分ずらす



# 【例題3-5】 1つだけLEDを点灯させて、上移動させてみよう。(rei3-5)

(7番目から順に0番目までの点灯するために、例題3-4の次の2か所を修正します。

※最初だけブロックは例題3-4と同じなので省略している。)

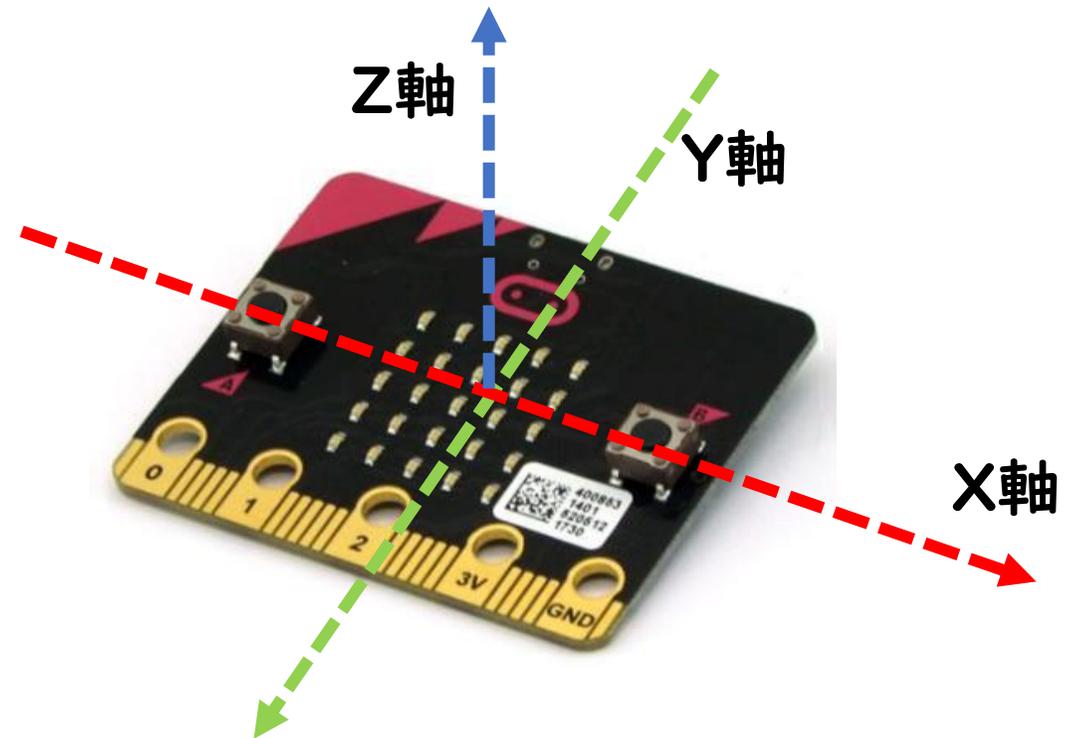


## 【コラム：加速度センサについて】

micro:bitには、3次元の加速度センサが搭載されており、図のようにx, y, zの3方向の加速度を計測することができます。

地上では下向きの重力加速度があるので、それを計測してmicro:bitを3次元的に動かすだけで値を変化させることができます。

また、動きや振動、衝撃なども検知することができます。



【例題3-6】 micro:bitを傾けて、搭載されている加速度センサから3つの値を読み込んでRGBの3つの色に対応したLEDを点灯させ、micro:bitを動かして色を変えてみよう。 (ファイル名:rei3-6)

<手順>

- 1) 「最初だけ」ブロックの中に「変数をNeoPixelモードにする」ブロックを追加する。  
-----
- 2) ずっとブロックの中に、NeoPixelのその他から「stripの0番目を赤色にする」ブロックを追加する。
- 3) 赤の部分を、NeoPixelのその他の「RGB(赤255、緑255、青255)」に置き換える。
- 4) 数値の255の部分を加速度X、加速度Y, 加速度Zに置き換える。
- 5) Neopixelから「Stripを指定した色で点灯する」ブロックを追加する。
- 6) 基本から一時停止(ミリ秒) 100を追加する。
- 7) NeoPixelから「stripに設定されている色を1個分ずらす」ブロックを追加する。
- 8) **作成したプログラムをmicro:bitに書き込みます。**

## 【例題3-6】 (ファイル名:rei3-6)

最初だけ

変数 `strip` を 端子 `P0` に接続しているLED 8個のNeoPixel (モード `RGB (GRB順)`) にする

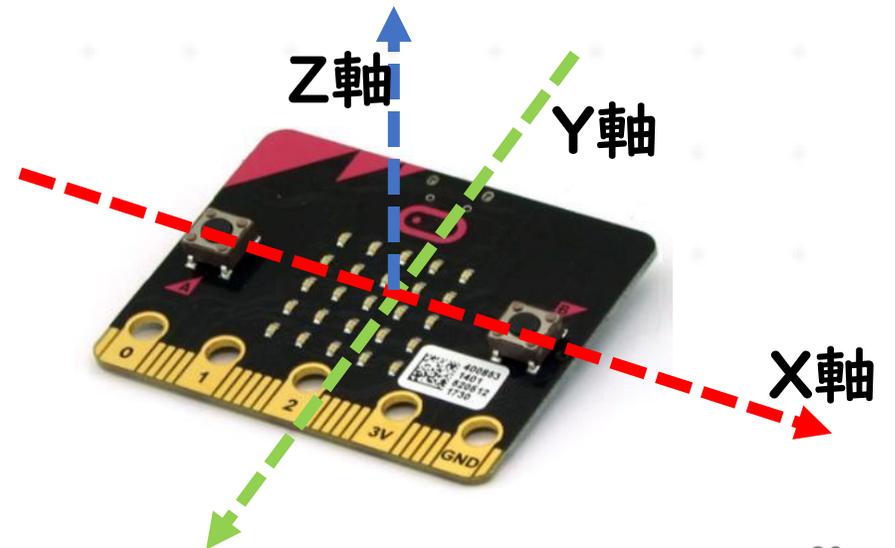
ずっと

`strip` の 0番目のLEDを RGB (赤 `加速度 X` 緑 `加速度 Y` 青 `加速度 Z`) 色に設定する

`strip` を設定した色で点灯する

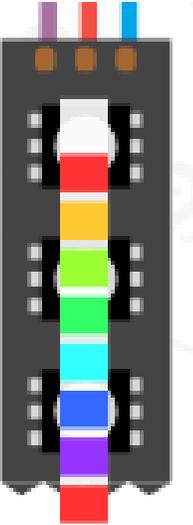
一時停止 (ミリ秒) 100

`strip` に設定されている色をLED 1個分ずらす



## 【例題3-7】 虹色で光らせてみよう。

NeoPixelのライブラリの中に「レインボーパターンで点灯する」という命令があり、それを使うと、虹色で光らせることができます。  
(ファイル名:rei3-7)



### <手順>

- 1) 「最初だけ」ブロックの中に、「変数をNeoPixelモードにする」ブロックを追加する。
- 2) 今回のNeoPixelの数は8個なので、ブロック内の24を8に変更する。
- 3) 次に「Stripをレインボーパターンに点灯する」ブロックを追加する。
- 4) **作成したプログラムをmicro:bitに書き込みます。**

最初だけ

変数  を 端子  に接続しているLED  個のNeoPixel (モード ) にする

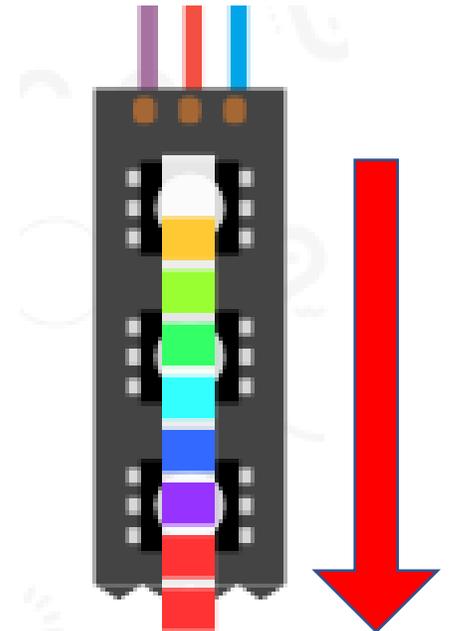
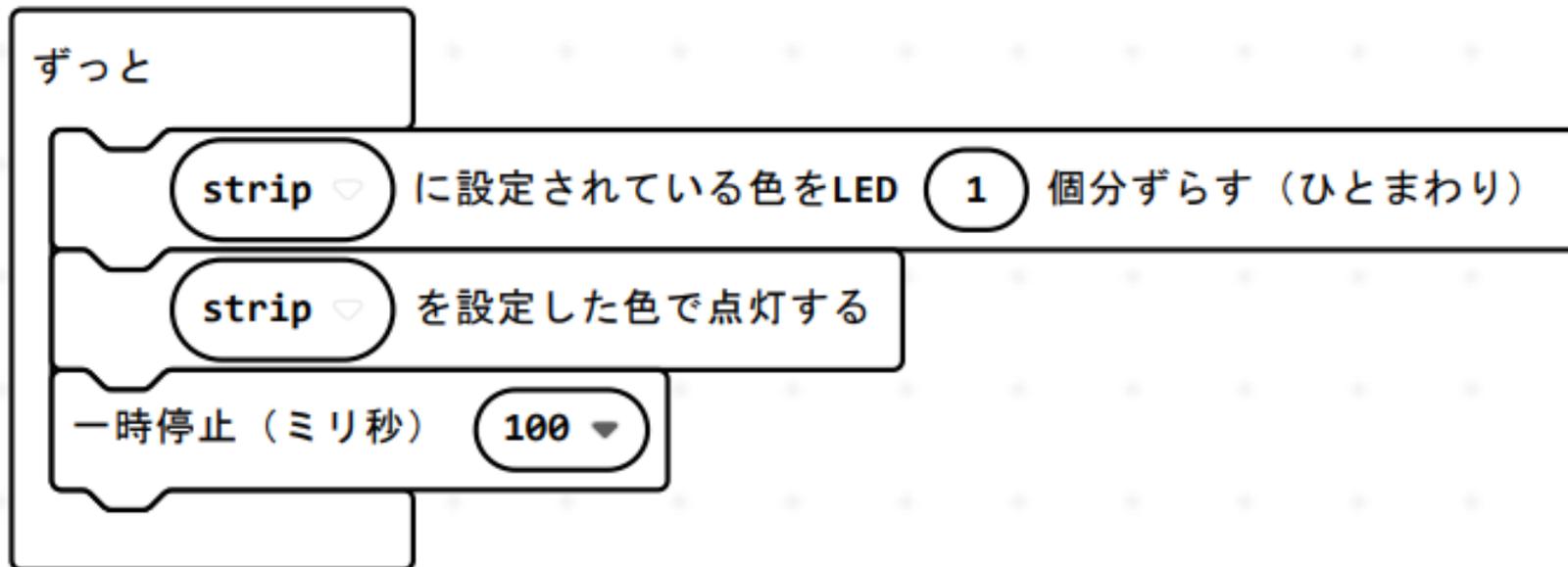
をレインボーパターン (色相  から ) に点灯する

【例題3-8】 レインボーパターンで下移動させよう。

例題3-7ではレインボーパターンの色が固定されていたので、「ずっと」ブロックの中に「1個分ずらす」ブロックを追加して、色が順番に下移動するようにしてみよう。（ファイル名:rei3-8）

<手順>

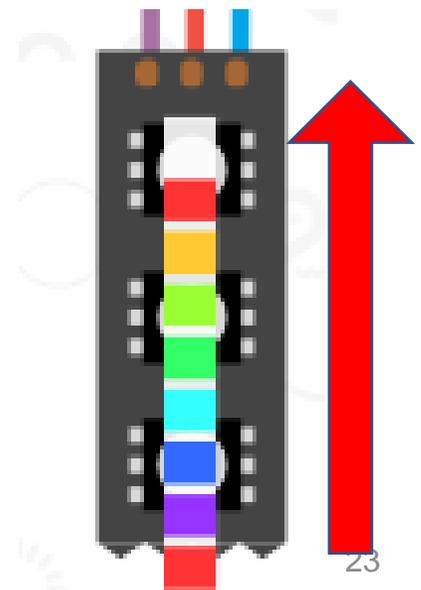
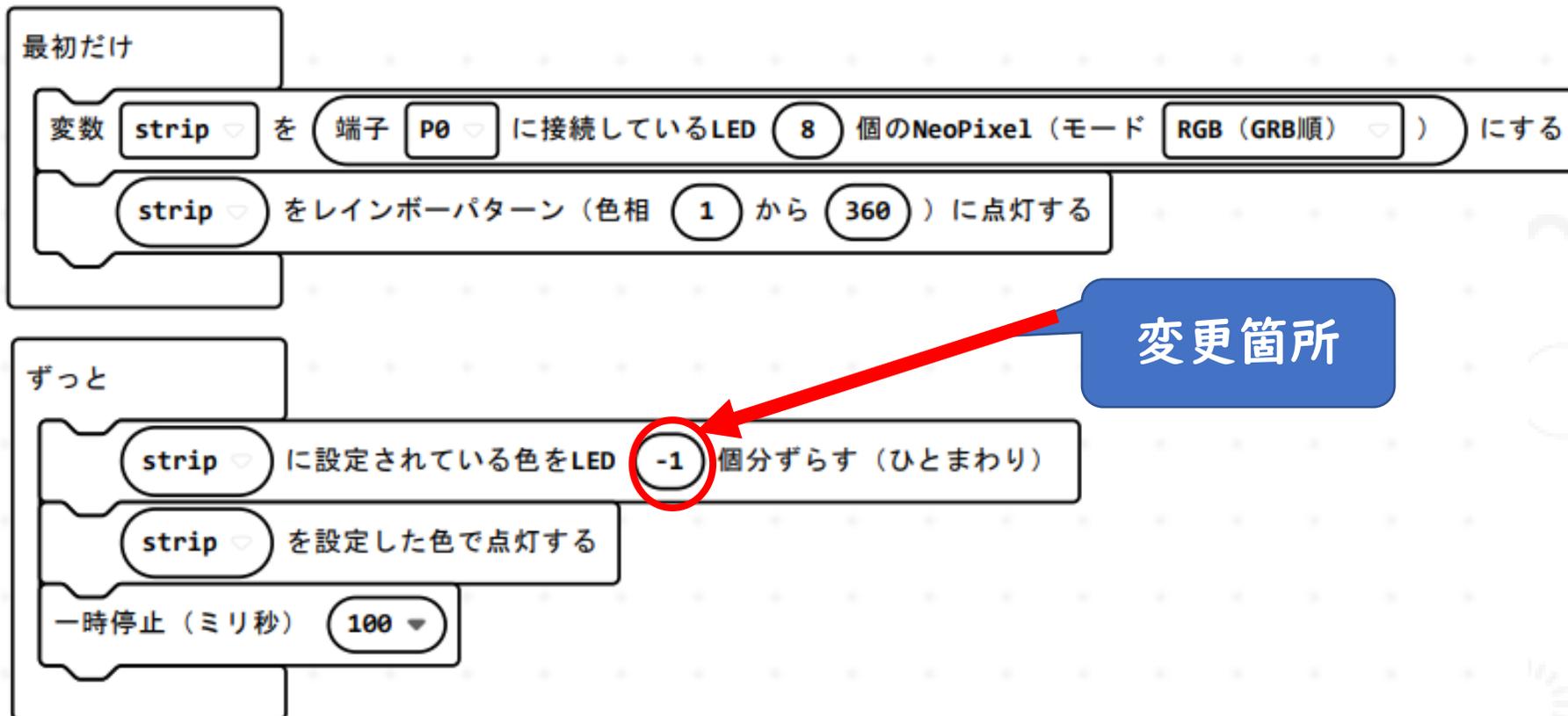
- 1) 「最初だけ」のブロックは、例題3-7をそのまま使います。
- 2) 「ずっと」ブロックの中に、Neopixelから「LED1個分ずらす（ひとまわり）」ブロックと「設定した色で点灯する」ブロックを追加する。
- 3) 基本から「一時停止（ミリ秒）100」ブロックを追加する。
- 4) 作成したプログラムをmicro:bitに書き込みます。



【例題3-9】 レインボーパターンで上移動させよう。  
例題3-8ではレインボーパターンで下移動したので、ここでは上移動する  
ようにしてみよう。 (ファイル名:rei3-9)

<手順>

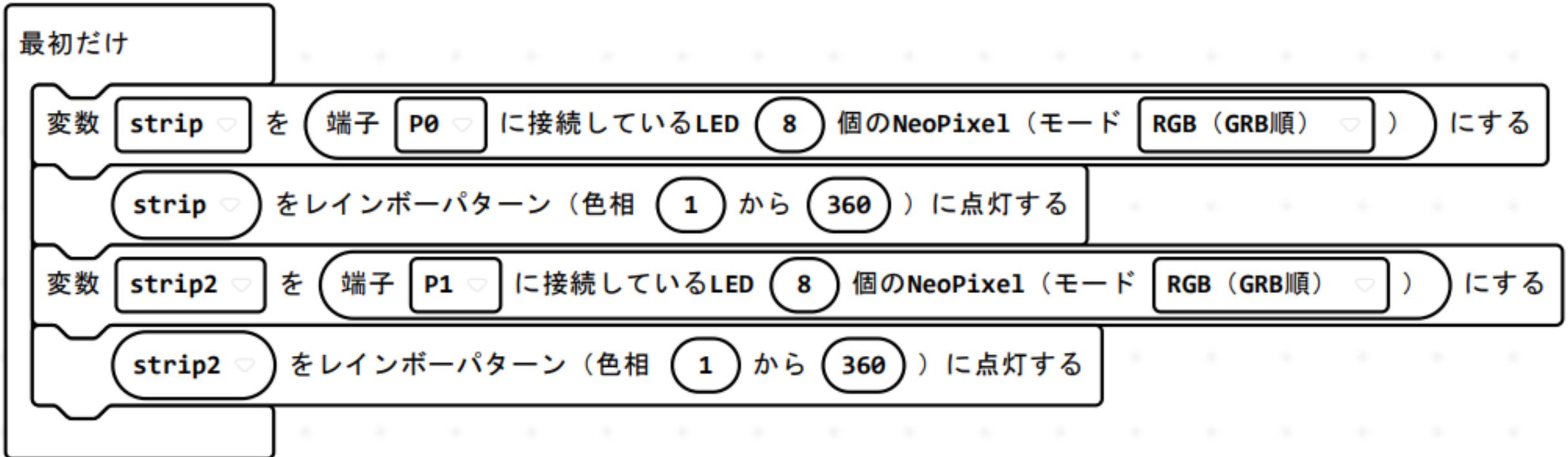
- 1) ブロックは、例題3-8を**基本的にそのまま使います**。
- 2) 「ずっと」ブロックの中の「LED 1個分ずらす (ひとまわり)」を「**-1**個分ずらす」  
に**変更**します。



【例題3-10】 2つのNeopixelを使って、レインボーパターンで、1つは下移動、もう1つは上移動で点灯させてみよう。（ファイル名:rei3-10）  
（1つ目はP0端子、2つ目はP1端子に接続するものとする）

<手順1>

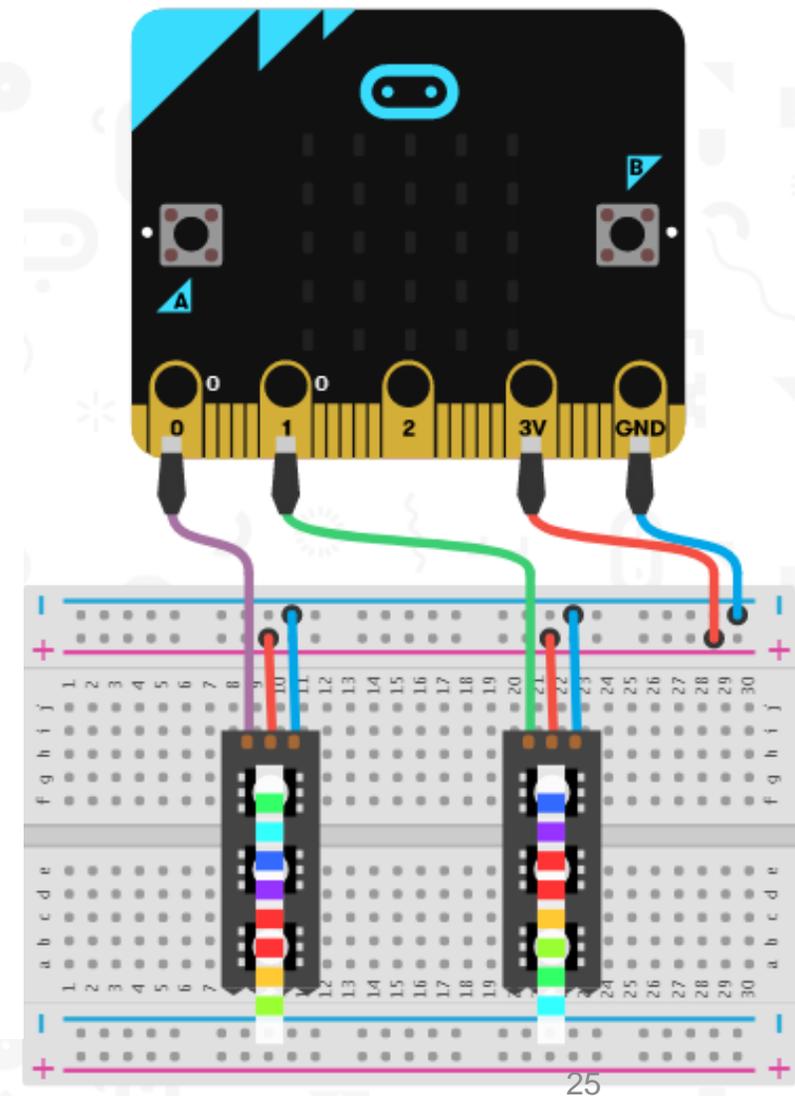
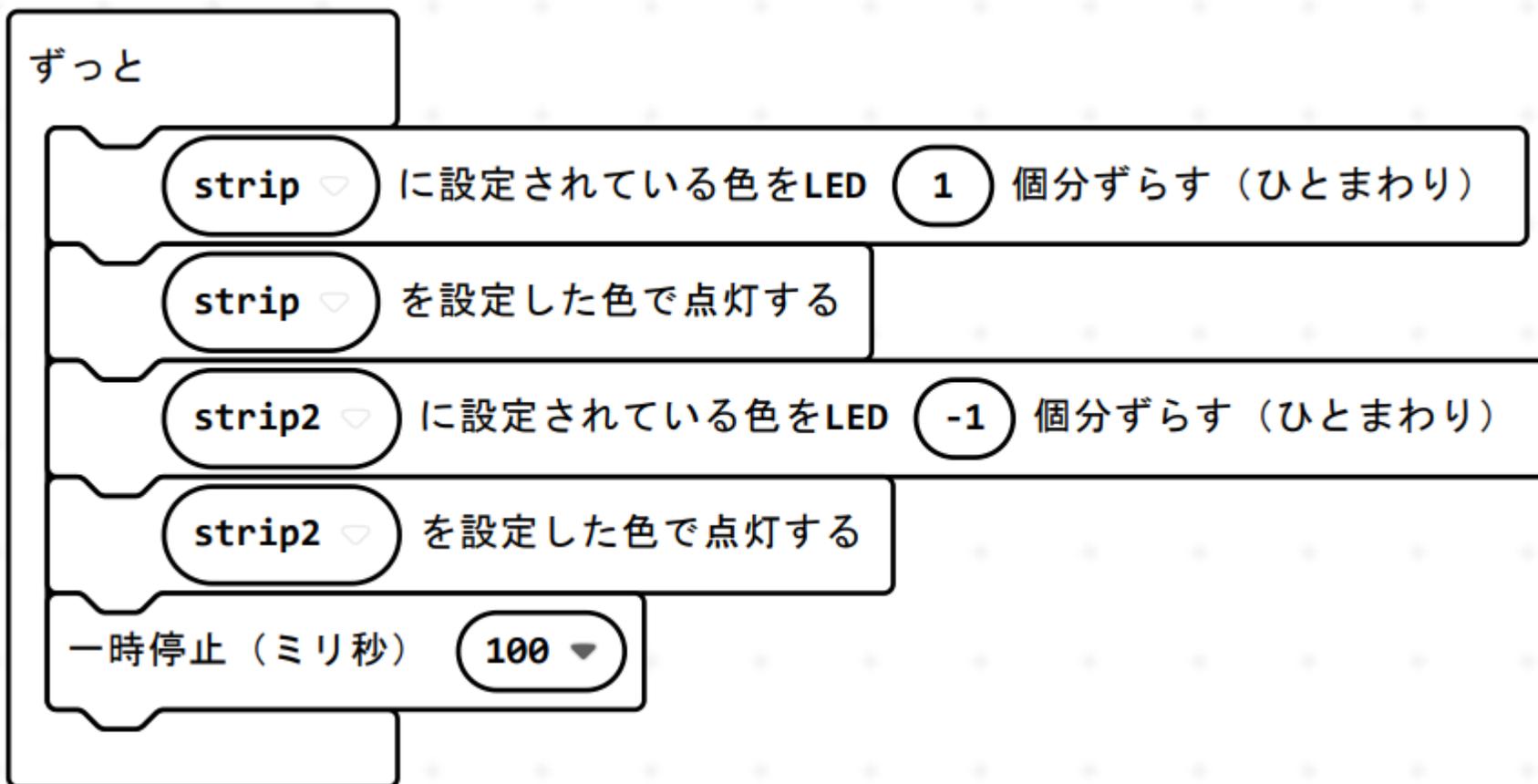
最初だけのブロックは、下の図のように2つの変数を使って、ブロックを作成する



## <手順2>

右図のように2つのNeopixelが表示されるので、ずっとブロックの中に、1つは下移動を、もう1つは上移動を設定する。

(ファイル名:rei3-10)



## プログラムのダウンロードについて

### <プログラムに対する注意事項>

本テキストで利用している例題プログラムなどは、NPO法人学習開発研究所の下記のWebサイトからダウンロードしてください。

本書の中で記載しているJavaScriptのファイル名、例えば、rei〇〇は、保存ファイル名では、microbit-rei〇〇.hex、 になっています。

<https://www.u-manabi.net/microbit/kensyu/>

