

データサイエンスに挑戦しよう！

～復習・応用編～

参加費 **無料!**
教材代 **500円** (希望者のみ)

今回の課題

共通: 前回のおさらい (40分)

・測定データの読み方、変化の観察 等

選択: 挑戦課題 (80分)

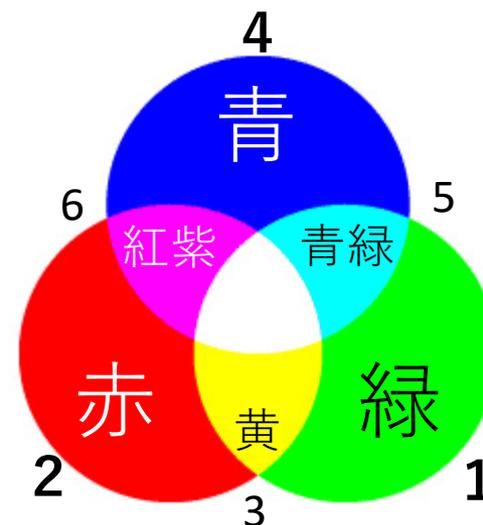
- ・ゲームづくり
- ・LEDを使ったカラー制御
- ・データの記録とグラフ表示



10個のカラーLEDを

- ・温度計
- ・イルミネーション
- ・ゲーム 等々

光で色づくりの基礎と応用



光の三原色 (RGB)

2026年2月22日 (日) 13時30分～15時30分 会場: 本願寺小樽別院

主催: NPO法人小樽青少年科学技術の芽を育てる会

協力: 浄土真宗本願寺派 本願寺小樽別院、NPO法人Nextday

コンピュータ (メモリー) に メモ (記憶) !

配列 ([0] ~ [101]) 102個と変数 (A~Z) 24個の計126個の数値を一時的に記憶できます。

変数 (配列+A~Z)

256

◆決められた3個の数字を配列に記憶

クリアバリアブル

```
CLV 変数、配列を全部0にする  
LET [0], 0, 1, 2
```

```
LET [3], 10, 11, 12
```

配列の数値を表示してみよう

```
?[0] → _____  
?[1] → _____  
?[2] → _____
```

```
?[3] → _____  
?[4] → _____  
?[5] → _____
```

- ・人間でも覚えることができる。ただし時間がかかる。

```
LET [0], 0, 1, 2, ▪ (連続30個) ▪, 28, 29
```

- ・人間でも覚えることができる？

```
LET [0], 4, 1, 7, ▪ (ランダム30個) ▪, 5, 10
```

チャレンジ



コンピューターに挑戦！

ランダムな数を記憶して表示

記憶して表示するまでの時間を競おう！！！！

LOAD 2

```
10 CLV:CLS:CLT
15 INPUT "? (Max101)",D
20 FOR I=1 TO D
30 LET [I],RND(100)
35 ? [I]
40 NEXT
45 BEEP50:IF !INKEY() CONT
47 ?
50 FOR J=1 TO D
60 ? "[ ";J;" ]=", [J]
70 NEXT
75 BEEP5
90 ? TICK()/60;"s"
```

IchigoJam (イチゴジャム) には、WS.LED命令という便利な機能があります。
これを使うと、光の三原色で約1677万色に光らせることができるんだ！

WS.LED

WS2812を光らせる命令

10

光らせるLED数



白い中に
3個のLED (RGB)
が入っている

配列 (記憶エリア) に設定された、色コードに従って光ります。

- ・ 1個のLEDで、R,G,Bの3つの値が必要
- ・ そのため配列を3つ使う

10個のLEDならば、30コ (0~29) の配列を使います。

	1	2	3	4	5	6	...	10
緑	: [0]	[3]	[6]	[]	[]	[]	...	[]
赤	: [1]	[4]	[7]	[]	[]	[]	...	[]
青	: [2]	[5]	[8]	[]	[]	[]	...	[]
配列番号 が増える数	3	3						



規則的に変化するため、
計算式で配列の番号を求める
ことができます

$$\text{緑色} = (\square \text{番目} - 1) \times 3$$

$$\text{赤色} = (\square \text{番目} - 1) \times 3 + 1$$

$$\text{青色} = (\square \text{番目} - 1) \times 3 + 2$$

LED色コード表

たくさんの色を作るのは難しいので、
8種類の色に限定して、とっても簡単に光らせてみよう！

色 code	[0] C&1*50	[1] C>>1&1*50	[2] C>>2&1*50	色	色配列
0	0	0	0	無	LET[0], , ,
1	5	0	0	緑	LET[0], 5, 0, 0
2	0	5	0	赤	LET[0], 0, 5, 0
3	5	5	0	黄	LET[0], , ,
4	0	0	5	青	LET[0], 0, 0, 5
5	5	0	5	青緑	LET[0], , ,
6	0	5	5	赤紫	LET[0], , ,
7	5	5	5	白	LET[0], 5, 5, 5

緑色に光らせる

緑

赤

青

LET[0], 50, 0, 0 : WS.LED1, 10

LET[3], 50, 0, 0 : WS.LED1, 10

LET[9], 50, 0, 0 : WS.LED1, 10

色を消す : CLV : WS.LED1, 10



色コード 変数C を使って光らせる

C = 7 <光らせたい色の番号 (0~7) に変える

```
LET [0], C & 1 * 5, C >> 1 & 1 * 5, C >> 2 & 1 * 5
```

```
WS.LED 1, 10
```

繰り返す回数 <10個のLEDが色コードの色に光る

光らせる回数

色コード(変数C)

- 0 無 (光らない)
- 1 緑
- 2 赤
- 3 黄
- 4 青
- 5 シアン (青緑)
- 6 マゼンタ (紅紫)
- 7 白

>>

「右シフト(Right Shift)」演算子

数値A >> 数値B

(数値Aを、右にBビット分シフトする)

? 8 >> 1

4

※8は2進数で 1000

右に1つずらして 0100=4

💡 明るさを変えてみよう

プログラムの中の「5」という数字が明るさを現わしています。

3色の明るさをそれぞれ変えて組み合わせることで、多くの色をつくります。

小さい数字 (例: 5) → 暗い

大きい数字 (例: 100) → 明るい

使える範囲: 0~255

IchigoJam&ESPJUCE用の、フルカラーLED温度計

毎回コマンド命令では大変なので、プログラムにして点灯させます。

```
^ FULL COLOR LED TEST
```

```
20 CLV
```

```
40 I=6          LED位置 0~9
```

```
41 C=7          LED色   1:緑 2:赤 3:黄 4:青 5:シアン 6:マゼンダ 7:白
```

```
42 L=100        LED明るさ0~255
```

```
110 LET[I*3],C&1*L,C>>1&1*L,C>>2&1*L
```

```
130 WS.LED 10
```

```

10 'ondokei-simple1
40 INPUT "18~27°",A
50 FOR I=0 TO 9
90 L=50 :温度計を点灯
100 IF I>A-18 L=0 '18度からLEDを点灯する、A度より上は点灯しない
110 LET[I*3],L,L,L
115 PRINT I+18,L
120 NEXT
130 WS.LED 10
140 GOTO 40

```

これでも良い

```

90 L=0
100 IF I<=A-18 L=50

```

もしくは、

```

100 IF I<=A-18 L=50 ELSE L=0

```

または、

```

100 IF I>A-18 L=0 ELSE L=50

```

温度Aの値を、LED白色で点灯する

温度表示を、任意の色で点灯する

```
10 ' ondokei-simple2
40 INPUT "18~27°",A
50 FOR I=0 TO 9
90   C=2:L=80           :温度計を赤色で点灯 C=1~7
100  IF I>A-18 C=0     :18度からLEDを点灯する
110  LET[I*3],C&1*L,C>>1&1*L,C>>2&1*L
115  ?I+18,C,L
120  NEXT:?
130  WS.LED 10
140  GOTO 40
```

```

10  'ondokei1
40  INPUT  "18~27°", A
50  FOR  I=0  TO  9
60  IF  I<2  C=4:L=99:GOTO  100      :'18~19° blue
70  IF  I<5  C=1:L=20:GOTO  100      :'20~22° green
80  IF  I<7  C=3:L=30:GOTO  100      :'23~24° yellow
90          C=2:L=80                  :'25~27° red
100  IF  I>A-18  C=0                  :'18度から温度計を表示する
110  LET[I*3], C&1*L, C>>1&1*L, C>>2&1*L
115  ?I, C, L
120  NEXT: ?
130  WS.LED  10
140  GOTO  40

```

温度の値によって、色を変える
低いほど青色
高いほど赤色

```

10 @ARUN'ondokei2
20 CLV:
30 @MAIN
40 'WAIT60:?'GETKIO":WAIT30
50 INPUT "8~27° ",A
100 IF A>17 GOTO@THEN ELSE GOTO@ELSE
110 @THEN          1 8度から2 7度の表示
120 T=18:S="5511133222":
130 GSB@ONDOKEI
140 GOTO@MAIN
150 @ELSE          8から1 7度の表示
160 T=8 :S="7744444555"
170 GSB@ONDOKEI
180 GOTO@MAIN
200 @ONDOKEI
210 Q=LEN(S):IF L==0 L=30
220 FOR I=0 TO Q-1
230 C=ASC(S+I)-48
240 IF I>A-T C=0
250 LET[I*3],C&1*L,C>>1&1*L,C>>2&1*L
260 ?I,C,L
270 NEXT:?
280 WS.LED Q
290 RTN

```

このプログラムは、温度に応じてLEDの色を変える電子温度計を作るものです。

温度(8~27度)を入力すると、
LEDテープが温度に合わせて色を変えます。

寒い時(8~17度) → 青っぽい色
暖かい時(18~27度) → 赤っぽい色

Tの温度を下限として、Sの1文字目からの色を順に設定する

@ONDOKEI で使用する変数

要設定変数

- A 温度
- T LED点灯させる下限の温度
- S 色設定 (LED数)
- L 輝度

内部使用

- Q LED個数 (変数Sから取得)
- I 点灯LED位置
- C 点灯LED位置の色

温度表示をサブルーチン化して、
2種類の温度範囲を色分けで点灯する

- 110 @THEN : 1 8度から2 7度の表示
- 150 @ELSE : 8から1 7度の表示

プログラムの流れ

スタート

↓

温度を入力してもらう

↓

18度より高い?

↓

YES

↓

NO

↓

暖かい色 寒い色

(赤系) (青系)

↓

↓

LEDを光らせる

↓

また温度入力へ戻る

色の数字の意味

各数字は**2進数で3つのLED(赤・緑・青)**を表しています:

数字	2進数	赤	緑	青	色
0	000	×	×	×	消灯
1	001	×	×	○	青
2	010	×	○	×	緑
3	011	×	○	○	シアン
4	100	○	×	×	赤
5	101	○	×	○	マゼンタ
6	110	○	○	×	黄色
7	111	○	○	○	白

各行の説明

10行目 @ARUN'ondokeyi2

プログラムの名前をつけています

「@ARUN」は自動実行の印

20行目 CLV:

画面をきれいにクリアします

30-40行目 @MAIN

メインプログラムの始まり

ここから繰り返し実行されます

50行目 INPUT "8~27° ",A

「8~27° 」と表示して、温度を入力してもらいます

入力した数字は変数Aに入ります

100行目 IF A>17 GOTO@THEN ELSE GOTO@ELSE

温度が18度より高いかチェック

高ければ@THEN(暖かい色)へ

低ければ@ELSE(寒い色)へ

 温度による色の違い

120行目(暖かい時) T=18:S="5511133222"

T=18: 18度から始まる

S="5511133222": 色の設定(数字が色を表す)

5→マゼンタ(紫)

1→青

3→シアン(水色)

2→緑

160行目(寒い時) T=8:S="7744444555"

T=8: 8度から始まる

S="7744444555": 色の設定

7→白

4→赤

5→マゼンタ(紫)

@ONDOKEIサブルーチン(200行目～)

これが実際にLEDを光らせる部分です。

210行目 Q=LEN(S):IF L==0 L=30

Q: 色の文字列の長さ(10個)

L: LEDの明るさ(初期値30)

220-270行目 FORループ

FOR I=0 TO Q-1 '0から9まで繰り返す

各LEDに対して:

230行目 C=ASC(S+I)-48

文字を数字に変換(例:"5"→5)

240行目 IF I>A-T C=0

温度より高い位置のLEDは消す

例: 温度20度で、T=18なら、I>2の位置は消灯

250行目 LET[I*3],C&1*L,C>>1&1*L,C>>2&1*L

LEDの色データを設定

C&1: 青のON/OFF

C>>1&1: 緑のON/OFF

C>>2&1: 赤のON/OFF

280行目 WS.LED Q

設定した色でLEDを実際に光らせる

290行目 RTN

サブルーチンを終了してメインに戻る

1. LED色の設定 `@SET_WS.LED(I,C,L)`

WS.LED命令を使うとき、色や動きを簡単に設定できるサブルーチンをまとめました。

WS.LED命令の概要は、「LEDでナナイロづくり」も参考にしてください。

SET_WS.LED	LED色の設定	https://nextday-kids.com/wp/975/1/
GET_WS.LED	LED色の取得	https://nextday-kids.com/wp/975/2/
STR_WS.LED	一括色設定	https://nextday-kids.com/wp/975/3/
SHIFT_WS.LED	LEDシフト設定	https://nextday-kids.com/wp/975/4/
LUMINA_WS.LED	輝度設定	https://nextday-kids.com/wp/975/5/



```
1000 @SET_WS.LED(I,C,L)
1010 IF I>33 THEN I=33
1020 IF (L<1||L>255)&&(C%8) THEN L=20
1030 LET[I*3],C&1*L,C>>1&1*L,C>>2&1*L
1040 RETURN
```

- ・ LED位置I番の配列（GRBの3か所）に、変数C（色）と変数L（明るさ）を設定します。

変数 I LED点灯位置

0 ~ 3 3 （LEDの数は最大3 4個まで）

変数 C LED色

0 ■ 無（光らない）、1 ■ 緑、2 ■ 赤、3 ■ 黄、4 ■ 青、
5 ■ シアン（青緑）、6 ■ マゼンタ（紅紫）、7 ■ 白

変数 L LED輝度

1 ~ 2 5 5 （範囲外は強制的に2 0）

※変数L=0は無効にしています。

LED光らせたくないときは、変数Cに0を設定してください。

もし、変数Lの0を有効にするときは、1020行目を削除してください。

※変数Cは下位3ビット分しか使用していません。C=9は、C=1と同じ色です。
マイナス値でも何かしら光ります（ $ABS(C\%8)$ と同じ）。

※変数Lも下位8ビットしか使用していません。

L=256はL=0と同じ明るさです（ $ABS(L\%256)$ と同じ）。

サンプルプログラム1 (7色を順に光らせる)

```
10 CLY
20 L=10
30 FOR C=1 TO 7
40 PRINT "-----"
50 FOR I=0 TO 11
60 GSB @SET_WS.LED(I,C,L)
70 PRINT "I=";I,"C=";C,"L=";L
80 WS.LED 12
90 WAIT 10
100 NEXT
110 ?
120 WAIT 60
130 NEXT
140 END
```

2. LED色の取得

@GET_WS.LED(I,C,L)

```
2000 @GET_WS.LED(I,C,L)
2010 IF I>33 THEN I=33
2020 C=( [ I*3 ] && 1 ) + ( [ I*3+1 ] && 1 ) * 2
                                     + ( [ I*3+2 ] && 1 ) * 4
2030 L = [ I*3 ] | [ I*3+1 ] | [ I*3+2 ]
2040 RETURN
```

- ・LED位置I番の、色と明るさを、変数Cと変数Lに取得します。
現在WS.LED命令で光っているLEDの値を知ることができます。
- ・ただし、@SET_WS.LEDサブルーチンで設定したRGB値でないと正しい値を取得できません。

変数 I

LED位置を指定する 0～33

変数 C

変数Iで指定したLED位置の色が返る。

0 ■ 無（光らない）、1 ■ 緑、2 ■ 赤、3 ■ 黄、4 ■ 青、
5 ■ シアン（青緑）、6 ■ マゼンタ（紅紫）、7 ■ 白

変数 L

変数Iで指定したLED位置の輝度が返る。 0～255

サンプルプログラム2

```
10  FOR I=0 TO 11
20  C=RND(8)
30  L=RND(256)
40  GSB @SET_WS.LED(I,C,L)
50  NEXT
60  WS.LED 12
70  WAIT 120
80  ? "I C L"
90  FOR I=0 TO 11
100  GSB @GET_WS.LED(I,C,L)
110  ? I,C,L
120  BEEP
130  WAIT 60
140  NEXT
150  END
```

3. 一括色設定

@STR_WS.LED(S,(I,C,L))

```
3000 @STR_WS.LED(S,(I,C,L))
3010 FOR I=0 TO LEN(S)-1
3020 C=ASC(S+I)-48
3030 GOSUB @SET_WS.LED(I,C,L)
3040 NEXT
3050 RETURN
```

- ・文字列を使用して、一度に複数個のLED色を設定します。

変数 S 色パターン

文字列でLED色0~7をLEDの個数分を指定します。

数字以外のエラーチェックはしていません。

IchigoJamの場合、35文字以上指定すると配列エラーになります。

変数 L 輝度

※変数 I 位置
変数 C 色

これらの変数は文字列の文字パターンを作るために、サブルーチン内で使います。
そのため、I,Cの値は、文字列最後の文字の設定値が入ります。

※このサブルーチンでは、@SET_WS.LEDサブルーチン呼びだします。
ただし、つぎのようにGOSUB命令を直接置き換えれば不要です。
各変数の範囲チェックはしません。

```
3030 LET [ I*3 ] , C&1*L , ( C&2&&1 ) *L , ( C&4&&1 ) *L
```

サンプルプログラム3

```
10 CLV  
20 S = " 123456700000 "  
30 L = 50  
40 GSB @STR_WS.LED ( S , ( I , C , L ) )  
50 WS.LED 12  
60 END
```

※**LEN(S)** 命令は文字列の長さを返します。

これを**WS.LED**命令の引数に使用すると、文字列の長さに応じた**LED**数を取得できます。

```
10 CLV
20 S="123456700000"
30 L=20+RND(40)
40 GSB @STR_WS.LED(S,(I,C,L))
50 WS.LED LEN(S)
60 ?STR$(S),LEN(S),L
70 END
```

4. LEDシフト設定

@SHIFT_WS.LED(R)

```
4000 @SHIFT_WS.LED(R)
4010 IF R<0 THEN GOTO @WSREV
4020 COPY#700,#800,R*6
      :COPY#806,#700,R*6
4030 LET[0],[R*3],[R*3+1],[R*3+2]
      :RETURN
4040 @WSREV:R=-R
4050 LET[R*3],[0],[1],[2]
4060 COPY#700,#806,R*6
      :COPY#800,#700,R*6:R=-R:RETURN
```

- ・ WS.LEDで点灯しているLED色を右または左に1つシフト（配列移動）します。
 - ・ Rが正数（+）だと右、負数（-）だと左にシフトします。
-

変数R シフト個数

整数（+プラス）なら右方向へ、R個分のLED色を1つずらす。

負数（-マイナス）なら左方向へ1つずらす。

※変数Rには通常はWS.LEDの数と同じ値を設定します。

最大シフトは32個までです（33以上はエラー）。

※シフトして空いたLED位置には、シフト前の変数R値の値を代入します。

つまりR=12のとき、右シフトとなり、12個目LEDの色が、1個目に入ります。

※ COPY命令の作業用メモリとして、PCGパターンのメモリエリアを使います。

このサブルーチンを使用した後にPCGパターンを使うときは、CLP命令で初期化してください（参照：COPY命令、CLP命令）。

サンプルプログラム4

```
20  CLV
30  S="123456700000"
40  L=50
50  GSB @STR_WS.LED(S,(I,C,L))
60  WS.LED LEN(S)
70  WAIT 60*3
75  R=LEN(S)
80  FOR M=1 TO 20
90  GSB @SHIFT_WS.LED(R)
100 WS.LED LEN(S)
110 WAIT 30
120 NEXT
130 WAIT 60*3
140 R=-R
150 GOTO 80
```

5. 輝度設定

@LUMINA_WS.LED(L,V,I)

```
5000 @LUMINA_WS.LED(L,V,I)
5010 IF V<1 OR V>34 OR !L THEN RETURN
5020 IF L>255 THEN L=255
5030 FOR I=0 TO V*3-1
5040 IF [I] THEN [I]=L
5050 NEXT
5060 RETURN
```

- ・ WS.LEDで点灯している輝度を変更します。

変数 L 変更する輝度 (1~255)

変数 V 変更するLED個数 (1~34)

いずれも範囲外の設定なら、サブルーチンは実行しない。

変数 I 位置

サブルーチン内部で使う作業用です。

サブルーチンから抜けた後は、変数V×3-1の値に変更されます。

サンプルプログラム5

```
10 CLV
20 S="123456700000"
30 L=75
40 GSB @SET_WS.LED(S,(I,C,L))
50 WS.LED LEN(S)
60 WAIT 120
70 FOR L=100 TO 0 STEP -5
80 V=LEN(S)
90 GSB @LUMINA_WS.LED(L,V,I)
100 WS.LED LEN(S)
110 WAIT 10
120 NEXT
130 END
```