

# プログラムの作り方の基本を学ぼう

2年 組 番 氏名

プログラムには基本の構造が3つあります。  
 これらを理解して応用すれば  
 どんなプログラムでも書けるようになります。

## プログラムの基本構造とは

プログラムは 「                   」 「                   」 「                   」 の3つの構造であらわされます。  
 全てのプログラムはこの3つの組み合わせで作ることができます。

## 「分岐」構造を含んだプログラムの学習

前は、「順次」構造のプログラムを学習しました。  
 しかし、これだけでは途中で \_\_\_\_\_ をしてプログラムの動作を変えることができません。  
 そこで、コンピュータ（ココナッツ）が今の状況を読み取って、それに応じた動作をするための「分岐」を含んだプログラムを学習してみましょう。

まずはスイッチの位置を読み取ってみます。  
 i? とするとスイッチの値を表示します。

```
COSMOS 仕様      i?          iの値(スイッチの位置 05,06,01,02 のいずれかを表示)
```

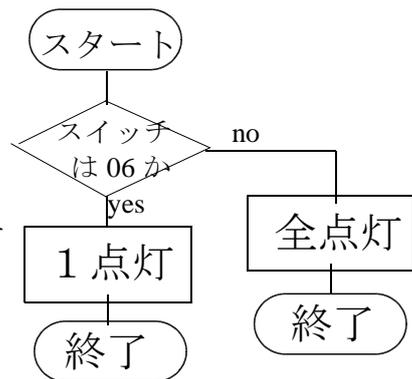
次に、値によって、実行する行を変化させる方法を確認します ★ここは特に難しいところ★  
 COSMOS 仕様 ?a=値 a が値に等しかったら次の行を実行、そうでなければ次の行を飛ばしてその次の行から実行する (ただし、a というのは様々な変数が入る)

— まとめ —

10	?i=05	< i が 05 なのかどうかを判断する命令 >	
20	命令文 1	【 i=05 だった場合にここを実行、その後は次の行へ順次実行 】	
30	命令文 2	【 i=05 ではなかった場合にここを実行 】	

□ 次のプログラムを入力して実行してみましょう。  
 プログラム名「スイッチの位置により  
 1つ点灯か全部点灯を選ぶ」

```
10 ?I=06          スイッチの判定
20 #=50          【 I が 06 だったら 】 50行へジャンプ
30 O=01          【 I が 06 でない場合】 1つ(0番)だけ点灯
40 .             プログラム終了
50 O=FF          I が 06 の場合 全部点灯
60 .             プログラム終了
```



これから I が 05 になる方を左スイッチ I が 06 になる方を右スイッチとすることにします。(間違っていたらスイッチを回してあわせておいてください)

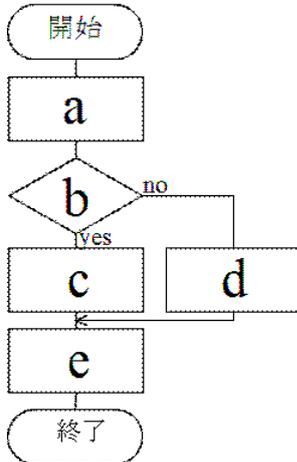
□ 練習課題  
 「切り替えスイッチの位置により、「1つおきに4つ点灯」と「2番が1つだけ点灯」を切り替えることができるプログラムを作りなさい。

~~~~~ フローチャートとは ~~~~~

今まで、プログラムの右側に図を使ってあらわしてきましたが、これをフローチャートといいます。プログラムの設計をするときに使われるもので、フローチャートに表すことによってプログラムの構造がよりわかりやすくなります。

命令や、構造により図の形が決められています。詳しくは別資料を見てください。

練習問題：生活での判断をフローチャートにしてみよう



乾電池を買いに電気店へ行って、A社とB社の乾電池のうち安い方を買って家に帰る。

- ① A社の乾電池を買う。
- ② B社の乾電池を買う。
- ③ 家に帰る。
- ④ 電気店に行く。
- ⑤ A社の乾電池の方が安い。

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| a : | b : | c : | d : | e : |
|-----|-----|-----|-----|-----|

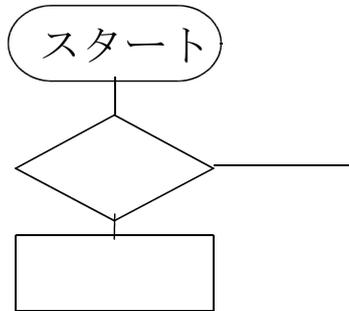
□練習問題

以下のプログラムをフローチャートで表しなさい。

また、そのプログラムをココナッツで作rinaさい。

「左側にスイッチを倒してスタートすると1番のLEDが点灯する。タクトスイッチを押しているときだけ全部のLEDが点灯する」

フローチャート(枠も途中で)



スイッチが左側の時にタクトスイッチを押すと、Iの値は05になります。

(書けたら先生を呼んで評価してもらおう)

ココナッツ  
プログラム  
できた?

今日の授業を振り返って

| 自己評価               |   |   |   |
|--------------------|---|---|---|
| 分岐実行するプログラムができたか   | ◎ | △ | × |
| スイッチの位置と値の関係がわかったか | ◎ | △ | × |
| フローチャートの書き方がわかったか  | ◎ | △ | × |

今日の授業の疑問点・感想を書こう

## ココナッツ & COSMOS(コスモス)の動作について



ココナッツの動作は以下の通りになります。

|                                    | パソコンと接続したとき                        | 接続しないとき          |
|------------------------------------|------------------------------------|------------------|
| 電源オン                               | プロンプトが出て入力待ち状態                     | 中に入っているプログラムを実行。 |
| 行番号 (0 1 ~ 9<br>5までの十進数)<br>をつけて入力 | プログラムエリアに格納 (実行はしない)               | /                |
| コマンドを入力                            | 直接実行する<br>例↓<br>R    格納されたプログラムを実行 |                  |

|         |                                         |
|---------|-----------------------------------------|
| R (行番号) | 指定した行番号からプログラムを実行                       |
| L       | プログラムのリストを表示                            |
| L (行番号) | 指定した行番号からプログラムのリストを表示                   |
| NA      | プログラム全体消去 (New All)                     |
| N (行番号) | 指定した行番号以降のプログラムを消去                      |
| MW      | メインメモリのプログラムをフラッシュメモリに保存 (電源を切っても保存される) |
| MR      | プログラムをフラッシュメモリからメインメモリに呼び出し             |
| CTRL+C  | プログラムを実行停止                              |
| CTRL+B  | スキップ実行                                  |
| CTRL+A  | 実行再開                                    |

|          |              |
|----------|--------------|
| (文字変数) ? | (文字変数) の値を表示 |
|----------|--------------|

### ■その他の仕様(きまり)と使用例

#### □定数

行番号は 10 進数 先頭から 2 桁 (T=xx,#=xx の代入等も)  
値は 16 進数 (00 ~ FF 正の整数のみ)

#### □代入 =

A=55

A=B

□演算 +,-,|,&^

A+55                    A=A+55  
A+B                     A=A+B  
A-55                    A=A-55  
A-B                     A=A-B

□判定 ?A=55                    真なら次行を実行

?A>55  
?A<55

□プログラムに関して

使えるコマンドや数値以外は全て無視をする仕様

行番号        01 ~ 95                    最大行数は 95 行  
存在する行番号のみ入力すると行の削除  
存在する行番号の入力時は上書き

・ジャンプ命令

10 #=50                    GOTO 50

・サブルーチン呼び出し命令

10 #:50                    CALL 50

・サブルーチン復帰命令

50 --- 実行したいプログラム ---

60 #:\$                    RETURN

・ブレークポイント

50 !

・終了

50 .

□レジスタ(文字の変数の決まり)

| 文字 | 機能                                                                           |
|----|------------------------------------------------------------------------------|
| A  | 汎用 (何にでも使える)                                                                 |
| B  | 汎用 (何にでも使える)                                                                 |
| C  | 汎用 (何にでも使える)                                                                 |
| D  | 汎用 (何にでも使える)                                                                 |
| E  | 汎用 (何にでも使える)                                                                 |
| F  | 汎用 (何にでも使える)                                                                 |
| G  | 汎用 (何にでも使える)                                                                 |
| O  | デジタル出力ポート        (ここに数字を入れると対応する LED が点灯)                                    |
| I  | 読み込んだとき: デジタル入力ポート: 現在のデジタル値(スイッチ位置)を取得<br>書き込んだとき: WAIT ( 1 を入れると 0. 1 秒待つ) |
| X  | 読み込んだとき: 外部アナログ入力 (AN1) の値を読み込む<br>書き込んだとき: PWM出力 (PWM ポートの制御)               |
| Y  | 読み込んだとき: 内蔵アナログ入力 (AN0) の値を読み込む<br>書き込んだとき: タイマー割り込み先 (0.1sec)               |
| #  | プログラムカウンタ (行番号を格納)                                                           |

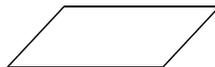
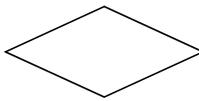
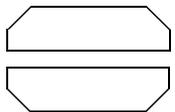
※内蔵アナログ入力は C D S (光センサー) に接続しています。

## フローチャートのきまりについて

フローチャート (flow chart) は処理の流れを簡単な図で整理して表すのに使います。

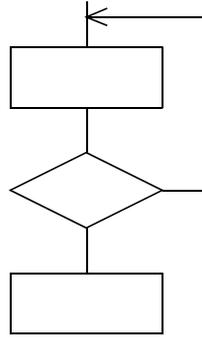
それぞれの部品は "JIS X 0121:1986" 「情報処理用流れ図・プログラム網図・システム資源図記号」により規格化されています。ここでは、ココナッツのプログラミングに使うであろう内容について抜粋して載せます。

### フローチャートで使う記号

| 記号                                                                                  | 名称     | 意味                                |
|-------------------------------------------------------------------------------------|--------|-----------------------------------|
| —————                                                                               | 流れ線    | 処理の順序を表す。順序を明確にするために、矢印をつけることもある。 |
|    | 端子     | 処理の開始・終了を表す。                      |
|  | 処理     | 一般的な処理を表す。                        |
|  | 準備     | 変数の宣言や初期値の設定などの処理を表す。             |
|  | 入出力    | データの入出力を表す。                       |
|  | 分岐     | 条件により、流れが2つ以上に分岐する処理を表す。          |
|  | ループ端   | 繰り返しの開始と終了を表す。                    |
|  | 定義済み処理 | 別に用意した処理 (サブルーチン) を利用することを表す。     |

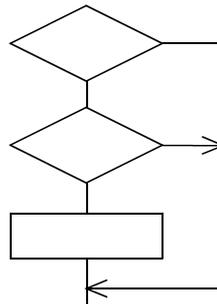
## フローチャートの書き方の決まり

- ① 処理の流れの方向は「左から右へ」、「上から下へ」とする。これ以外の場合は、矢印を使って明示する。



- ② 流れ線は互いに交差してもよい

- ③ 2つ以上の流れ線を集めて、1つにしてもよい。



## フローチャートの基本的な書き方例

