

# 英語風 Sunaba 早見

## 1 メモリ変更行

```
memory[番号を計算する計算式や数] -> 計算式や数
名前付きメモリの名前 -> 計算式や数
```

「->」の左に何番のメモリに覚えさせるかを指定し、右には、覚えさせる数を計算する計算式や、数を書く。メモリの番号指定をする代わりに、後述の名前付きメモリを使っても良い。例えば、

```
memory[3] -> 5
```

で、3番のメモリが5を覚える。

## 2 計算

計算式は、数、名前付きメモリ (後述)、メモリなどを計算記号 (演算子) でつないだものだ。例えば、 $2 + 3$ 、 $a * memory[4]$  のように書ける。記号は以下の 10 個である。

| 記号     | +  | -   | *  | /  | <      | >      | <=     | >=     | =      | !=     |
|--------|----|-----|----|----|--------|--------|--------|--------|--------|--------|
| 作用     | 加算 | 減算  | 乗算 | 除算 | 左<右で 1 | 左>右で 1 | 左≤右で 1 | 左≥右で 1 | 左=右で 1 | 左≠右で 1 |
| -7 と 5 | -2 | -12 | 35 | -1 | 1      | 0      | 1      | 0      | 0      | 1      |

後ろ 6 つの記号は 1 でない時は 0 になる。例えば  $4 > 6$  は 4 が 6 より大きくないので 0 になる。

計算は全て左から順に行う。 $4 + 5 * 6$  は  $4 + 5$  を先にやるので 54 になる。乗算や除算を先にやることはなく、優先順位は括弧で示す。 $4 + (5 * 6)$  と書けば 34 になる。

## 3 繰り返しと条件実行

プログラムは基本的には、1 行づつ上から下へと実行されるが、その流れを変える方法もある。それが以下の二つだ。

```
while 計算式や数
  中身
if 計算式や数
  中身
```

while は、計算式の計算結果や数が 0 でない間、そこに続く「中身」、つまり先頭に空白が空いた行 (複数行あってもいい) をを繰り返し実行する。複数行ある場合先頭の空白は同じ数である必要がある。また、if は、最大繰り返し回数が 1 の while である。

そしてこれらは多重にもできる。例えば、

```
while 計算式や数
  if 計算式や数
    if の中身
  while の中身
```

## 4 名前付きメモリ

名前付きメモリとはある番号のメモリの別名である。

a -> 5

と書くと、どこかの番号のメモリに a という名前がつき、それに 5 を覚えさせることができる。番号は気にしないでいい。作った名前付きメモリは計算に使える。

memory[4] -> a + 5

## 5 部分プログラム

部分プログラムはプログラムの一部に名前をつける仕組みである。

余りを出す(a, b) とは  
出力 -> (a - ((a / b) \* b))

と書くと「余りを出す」という名前の部分プログラムができる。a と b は「入力」という特別な名前付きメモリであり、「出力」もまた特別な名前付きメモリである。作った部分プログラムを使うには例えば以下のようにする。

答え -> 余りを出す(16, 5)

「出力」に覚えさせた数が「余りを出す(16, 5)」を置き換え、名前付きメモリ「答え」は 1 を覚える。部分プログラムは使われた時に初めて実行される。

「出力」に何も覚えさせない部分プログラムも作ることができ、その場合は、

点を描く(16, 5)

のように「->」なしで使う。

また、部分プログラムの中では外の名前付きメモリを使えない。部分プログラムの外では中の名前付きメモリは使えない。

## 6 メモリの番号

| 番号の範囲       | 説明  |
|-------------|---|
| 0-39999     | 自由領域 + プログラム                                  |
| 50000-50001 | マウスカーソル座標 (順に X,Y)。                           |
| 50002-50003 | マウスボタン (順に左、右。on が 1、off が 0)                 |
| 50004-50009 | キーボード (順に上、下、左、右、スペース、エンター。全て on が 1、off が 0) |
| 55000       | 同期スイッチ (画面反映)。何かを書き込むと画面にメモリを反映させる。           |
| 55001       | 自動同期無効化。1 で無効、0 で有効                           |
| 60000-69999 | 画素の色を格納。                                      |

### 6.1 画面 (60000 番台) について

幅が 100、高さが 100 で、左上から右へ、端まで行ったら下へ 1 段ずれる、という順番。

|      |      |      |       |      |      |      |
|------|------|------|-------|------|------|------|
| 0    | 1    | 2    | ..... | 97   | 98   | 99   |
| 100  | 101  | 102  | ..... | 197  | 198  | 199  |
| .    | .    | .    |       | .    | .    | .    |
| 9900 | 9901 | 9902 | ..... | 9997 | 9998 | 9999 |

この表に 60000 を加えた番号を使う。

色は 0 から 999999 まで。6 桁の数値の、上 2 桁が赤、次の 2 桁が緑、下の 2 桁が青。光の三原色で表す。例えば、

|            |        |        |               |             |        |               |        |
|------------|--------|--------|---------------|-------------|--------|---------------|--------|
| 黒          | 白      | 赤      | 緑             | 青           | 黄      | 空色            | 紫      |
| 000000(=0) | 999999 | 990000 | 009900(=9900) | 000099(=99) | 999900 | 009999(=9999) | 990099 |