

## NScripter の応用 (Ver.2)

アドベンチャーゲームなどを作れるソフト「NScripter」(エヌスクリプター)の応用です。

### ●表示効果

背景やキャラクターの立ち絵の画像を表示する時、これまではパッと一瞬で表示していました。これを「だんだん現れる」などの効果をつけて表示することができます。

### ★表示効果の番号を設定する

テストプログラムに、表示効果をつけてみましょう。

```
*define
caption "テスト"
deletemenu
game

*start
setwindow 16, 350, 25, 4, 24, 24, 0, 6, 20, 1, 1, #999999, 8, 345, 631, 471
bg "img¥haikei. jpg", 1
ld l, "img¥sarutobi. bmp", 1
ぼく、SARUTOBI くん¥
cl l, 1
end
```

表示効果を使うには、まず初期設定部分(「\*define」～「game」の間)で、「effect」(エフェクト)命令で呼び出し番号を設定します。

例えば、以下の設定を入れてみてください。

```
*define
caption "テスト"
deletemenu
effect 2, 10, 500
game
```

effect 2, 10, 500

効果呼び出し  
番号

エフェクト番号  
10 はクロスフェード  
(だんだん現れる)

効果秒数(ミリ秒)  
1000 分の1秒単位  
1000 ミリ秒=1秒

※呼び出し番号は、「1」は瞬間表示に使われているので、「2」より大きい数字をつけます。

## ★背景や立ち絵の表示に効果をつける

「effect」命令で設定ができれば、背景表示に効果をつけてみましょう。

背景を表示する「bg」命令の効果呼び出し番号を「2」にします。

ファイルを保存して、NScripter を実行してみてください。

背景がゆっくり表示されましたか？

```
bg "img¥haikei.jpg", 2
```

同じように、キャラクターの立ち絵の表示にも、効果をつけてみましょう。

立ち絵を表示する「ld」命令の効果呼び出し番号を「2」にします。

ファイルを保存して、NScripter を実行してみてください。

キャラクターがゆっくり表示されましたか？

```
ld 1, "img¥sarutobi.bmp", 2
```



キャラクターの立ち絵を消す時にも、効果がつけられます。

立ち絵を消す「cl」命令の効果呼び出し番号を「2」にします。

ファイルを保存して、NScripter を実行してみてください。

キャラクターがゆっくり消えましたか？

```
cl 1, 2
```

このように、効果呼び出し番号を最初に設定すると、何度もその番号で呼び出すことができます。いろいろな表示効果を「effect」命令で設定して、呼び出してみてください。

(例)

```
effect 3, 14, 500 ;呼び出し番号 3 を設定、効果=下からスクロール
```

```
effect 4, 4, 500 ;呼び出し番号 4 を設定、効果=上シャッター
```

```
bg "img¥haikei.jpg", 3
```

```
ld 1, "img¥sarutobi.bmp", 4
```

「effect」命令の2番目の数値で指定する「エフェクト番号」は、以下の 0～18 があります。

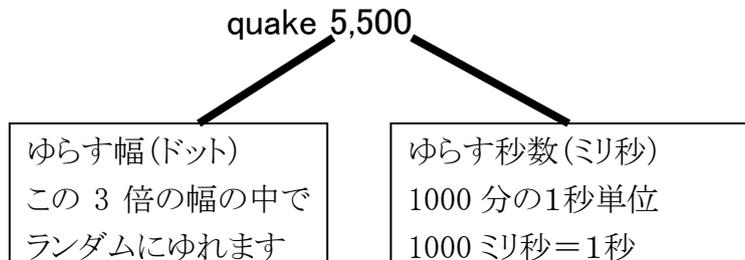
番号	効果
0	表示効果なし。標準で呼び出し番号「0」に設定されています。 0番をつけた画像は、その場では表示されません。 次に0番以外の番号をつけた画像を表示すると、一緒に表示されます。 立ち絵を左右2枚同時に表示したい時などに使います。
1	瞬間表示。標準で呼び出し番号「1」に設定されています。
2	左シャッター。
3	右シャッター。
4	上シャッター。
5	下シャッター。 2～5のシャッター系は、画像を複数のたんざく形にカットして表示します。 それぞれ左右上下方向に向けて表示されます。
6	左カーテン。
7	右カーテン。
8	上カーテン。
9	下カーテン。 6～9のカーテン系は、シャッターよりも大き目のたんざく形にカットして表示します。
10	クロスフェード。透明→不透明にだんだんと表示します。
11	左からスクロール。
12	右からスクロール。
13	上からスクロール。
14	下からスクロール。 11～14のスクロール系は、画面全体がスクロールする効果です。 ただし、背景も立ち絵も一緒にスクロールしてしまうので、背景だけが表示されている時、色背景+立ち絵の時などに使うといいでしょう。 色背景+立ち絵で、立ち絵表示の「ld」命令にこの効果をつけると、キャラクターが動いて登場するようになります。
15	マスクパターンによるフェード。難しいので省略。
16	モザイクアウト。画像がだんだん大きなモザイクに変化します。背景も立ち絵も一緒にモザイクになります。 「cl」命令で使うと、キャラクターがおかしな消え方をするように見えます。
17	モザイクイン。画像がだんだん小さなモザイクになって表示されます。背景も立ち絵も一緒にモザイクになります。 「bg」や「ld」命令で使うと、だんだんレンズの焦点が合う、キャラクターがワープして出現する、などの表現ができます。
18	マスクパターンによるクロスフェード。難しいので省略。

## ●画面をゆらす

表示されている画面をゆらすことができます。

キャラクターが何かにぶつかった、バトルで攻撃を受けた、などの効果として使えます。

画面をゆらすには「quake」(クエイク)という命令を使います。



SARUTOBi さんの立ち絵を表示した後に、画面をゆらしてみましよう。

SARUTOBi さんが何かにぶつかったような効果がつけられます。

ゆらす幅や時間をいろいろ変えて、試してみてください。

```
Id 1, "img¥sarutobi.bmp", 2
ぼく、SARUTOBi くん¥
quake 5, 500
いてて、ぶつかった¥
```

「quake」命令は上下左右ランダムにゆれますが、左右方向だけ、あるいは上下方向だけゆらすこともできます。

**quakex** 回数,時間 ←指定した回数と時間だけ左右にゆれる

**quakey** 回数,時間 ←指定した回数と時間だけ上下にゆれる

例えば、「quakey」命令で上下に1回ゆらすとキャラクターがうなずいたように(Yes)、「quakex」命令で左右に1回ゆらすとキャラクターが首を振ったように(No)見えます。

回数や時間をいろいろ変えて、試してみてください。

```
Id 1, "img¥sarutobi.bmp", 2
ぼく、SARUTOBi くん¥
quakey 1, 1000
うん、わかった¥
quakex 1, 1000
ううん、そうじゃない¥
```

**●音を出す**

BGM や効果音などの音を出す方法です。音が出ると、ゲームにとっても臨場感が出ます。

**★BGM を流す**

「bgm」(बीジーエム) 命令を使って、ゲームの背景に流れる音楽を設定します。

`bgm "ファイル名"`

BGM のファイルは、「bgm」フォルダに入っています。  
ゲームがスタートする所で、「bgm」命令を入れてみましょう。

```
*start
setwindow
16, 350, 25, 4, 24, 24, 0, 6, 20, 1, 1, #999999, 8, 345, 631, 471
bgm "bgm¥bgm01. wav" ←
bg "img¥haikai. jpg", 1
```

「bgm」命令は、曲が終わるとまた最初からくり返して流れます。(ループ再生)  
1 回だけ流して終わりにしたい時は、「bgmonce」(ビージーエム・ワンス) 命令を使います。

`bgmonce "ファイル名"`

流れている BGM を止めるには、「bgmstop」(ビージーエム・ストップ) 命令を使います。

`bgmstop`

BGM の音量(ボリューム) 調整は、「bgmvol」(ビージーエム・ボリューム) 命令を使います。

`bgmvol` 音量(0~100) ←0が最小、100 が最大

BGM の音量が大きいと、他の効果音やせりふが聞きづらいので、音量は 50 くらいにした方がいいでしょう。

BGM を変えたり、命令を変えたりして、いろいろ試してみてください。

## ★効果音を出す

キャラクターの動き、場面転換などに合わせて、効果音を出すことができます。  
効果音を出すには、「dwave」(ディーウェーブ)命令を使います。

**dwave** チャンネル番号, "ファイル名"

「チャンネル番号」は 0~49 の数字を指定します。  
キャラクターのせりふはチャンネル 0、それ以外の効果音はチャンネル 1~49 にします。

効果音のファイルは、「sound」フォルダに入っています。

SARUTOBi くんが登場する所で、「dwave」命令を入れて、効果音をつけてみましょう。

```
dwave 1, "sound¥sound01.wav" ←
ld l, "img¥sarutobi.bmp", 1
ぼく、SARUTOBi くん¥
```

「dwave」命令は、1 回だけ効果音を再生します。  
くり返し再生したい時は、「dwaveloop」(ディーウェーブ・ループ)命令を使います。

**dwaveloop** チャンネル番号, "ファイル名"

効果音を途中で止めたい時は、「dwavestop」(ディーウェーブ・ストップ)命令を使います。

**dwavestop** チャンネル番号

チャンネルを変えると、2つ以上の効果音を同時に再生できます。

```
dwave 1, "sound¥sound01.wav"
dwave 2, "sound¥sound02.wav"
```

この場合、2 つの効果音が重なって聞こえます。

1 番目の効果音が終わってから 2 番目の効果音を出したい時は、間にクリック待ちのセリフをはさむといいでしょう。

```
dwave 1, "sound¥sound01.wav"
ぼく、SARUTOBi くん¥           ;←クリック待ちの間に 1 番目の効果音が終わる
dwave 2, "sound¥sound02.wav"
```

## ●変数を使う

例えば、バトルゲームのプログラムで、最初にプレイヤーの名前やレベル、体力などを設定して、その後の場面で使うことがあります。

これを実現するには、数値や文字を記憶しておく「変数」(へんすう)を使います。変数は、数値や文字を入れる箱のようなものです。

まず、プレイヤーの名前と体力(HP)を設定してみましょう。

ゲームが開始する時に、変数の値を設定します。

### ★文字変数

文字が入る「文字変数」は、「\$」に続けて数字を書きます。

文字変数「\$1」を作って、プレイヤーの名前を設定してみましょう。

変数に値をセットするには「mov」(ムーブ)命令を使います。



指定する文字は、「"」(ダブルクォーテーション)で囲みます。

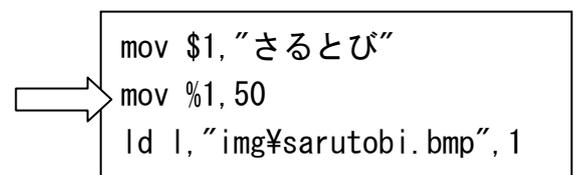
### ★数値変数

数値が入る「数値変数」は、「%」に続けて数字を書きます。

数値変数「%1」を作って、プレイヤーの体力を設定してみましょう。

mov %1,50

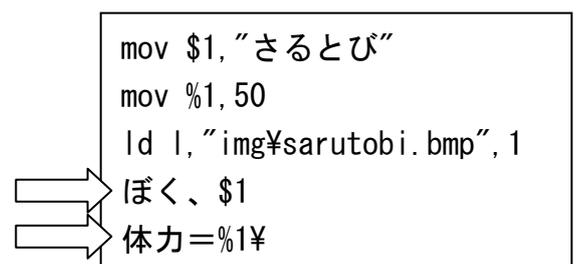
なお、変数の番号は、0~199 までが使えます。



変数に設定した名前と体力を、画面に表示してみましょう。

表示する文章の中に変数名を書くと、一緒に表示されます。

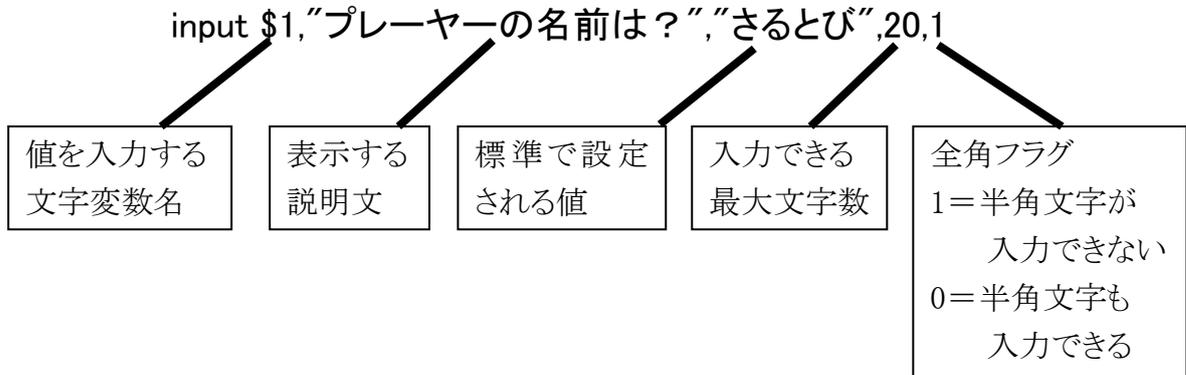
名前や体力の値をいろいろ変えて、試してみてください。



## ★文字を入力する

プレイヤーの名前を、キーボードから入力できるようにしてみましょう。

「mov」命令で文字変数「\$1」を指定している行を削除して、代わりに「input」命令を書きます。



```
input $1,"プレイヤーの名前は?","さるとび",20,1
mov %1,50
ld l,"img¥sarutobi.bmp",1
```

実行すると、文字列を入力するためのウインドウが開きます。



名前を入力して「決定」ボタンを押すと、その名前がプレイヤーの名前として表示されます。いろいろな名前を入れてみてください。

なお、「input」命令の最後の「全角フラグ」を「0」にして、半角文字も入力できるようにすると、名前を画面に表示した時に表示がおかしくなります。通常は「1」にして、半角文字が入力できないようにしてください。

## ★数値を入力する

次にプレイヤーの体力の値を、キーボードから入力できるようにしてみましょう。

「mov」命令で数値変数「%1」を指定している行を削除して、代わりに「input」命令を書きます。

```
input $2,"体力は？(半角文字で)","50",10,0
```

「input」命令は文字変数にしか入力できないので、新しく文字変数「\$2」を作って、そこに金額を入力します。数字なので、半角文字で入力するようにします。

そして、文字変数「\$2」の値を、数値変数「%1」へ変換します。

変換には「atoi」(エー・トゥー・アイ)という命令を使います。

```
atoi %1,$2 ;文字変数$2に入っている数字を変換して%1へ入れる
```

```
input $1,"プレイヤーの名前は？","さるとび",20,1
input $2,"体力は？(半角文字で)","50",10,0
atoi %1,$2
ld l,"img¥sarutobi.bmp",1
```

実行すると、名前の次に、体力の入力ウインドウが表示されます。



半角の数字を入力すると、その数値が体力として表示されます。

## ●バトルゲームを作る

体力の数値変数を応用して、バトルゲームを作ってみましょう。  
ここでは、SARUTOBi くとタヌキおやじをバトルさせることにします。

SARUTOBi くの次に、タヌキおやじを登場させます。  
タヌキおやじの体力は変数%2にして、値は30に設定しましょう。

```
ld r, "img\tanuki.bmp", 1
mov %2, 30
わしがタヌキおやじだ！
体力=%2¥
```

次にバトルが始まって、まず SARUTOBi くんがタヌキおやじへ攻撃します。

```
バトル開始！¥
$1 の攻撃！¥
```

### ★数値変数の計算

SARUTOBi くんが攻撃すると、相手のタヌキおやじの体力が減らされます。  
とりあえず体力を 10 減らすことにします。  
タヌキおやじの体力の変数%2 から 10 を引き算します。  
先にやった「mov」命令で、引き算ができます。

```
mov %2,%2-10 ;%2 の数値から 10 を引いて%2 へ入れる
```

これを入れて、タヌキおやじの体力が減らされる様子を表示してみましょう。

```
mov %2, %2-10
タヌキおやじに 10 のダメージ！
体力=%2¥
```

同じように、今度はタヌキおやじが SARUTOBi くんへ攻撃します。  
SARUTOBi くんの変数%1 を 10 減らします。

```
タヌキおやじの攻撃！¥
mov %1, %1-10
$1 に 10 のダメージ！
体力=%1¥
```

数値変数では、足し算、引き算、かけ算、割り算などの計算ができます。

```
mov %1,%1+10    ;%1 の数値に 10 を足して%1 へ入れる
mov %1,%1-10    ;%1 の数値から 10 を引いて%1 へ入れる
mov %1,%1*10    ;%1 の数値に 10 をかけて%1 へ入れる
mov %1,%1/10    ;%1 の数値を 10 で割って%1 へ入れる
```

なお、それぞれの計算をする専用の命令もあります。

```
add %1,10      ;(アッド) %1 の数値に 10 を足して%1 へ入れる
sub %1,10      ;(サブ、subtract の略) %1 の数値から 10 を引いて%1 へ入れる
mul %1,10      ;(マル、multiply の略) %1 の数値に 10 をかけて%1 へ入れる
div %1,10      ;(ディブ、divide の略) %1 の数値を 10 で割って%1 へ入れる
```

計算の行を書き換えて、いろいろな計算を試してみてください。

### ★数値変数で条件分岐(じょうけんぶんき)

バトルでのそれぞれの攻撃の後、残った体力の値で勝ち・負けを判断して、ストーリーを変えてみましょう。(もし～なら～、「条件分岐」)

条件は以下のとおりです。

- SARUTOBi さんの攻撃後、タヌキおやじの体力%2 が 0 以下になったら、SARUTOBi さんの勝ち
- タヌキおやじの攻撃後、SARUTOBi さんの体力%1 が 0 以下になったら、タヌキおやじの勝ち
- それ以外の場合は、バトルを続ける

条件分岐には「if」(イフ)命令を使います。



この例は、「変数%2 が 0 以下の時は、ラベル\*sarutobikati へジャンプする」という意味になります。

右に書いたプログラムを見て下さい。  
バトルが開始した後に、新しいラベル「\*battle」  
を入れています。後でここへ戻るためです。

if 命令を入れて、条件分岐をします。  
SARUTOBi くんが攻撃した後、タヌキおやじの  
体力%2 が 0 以下だったら、ラベル\*sarutobikati  
へジャンプします。

同じように、タヌキおやじが攻撃した後、  
SARUTOBi くんの体力%1 が 0 以下だったら、ラ  
ベル\*tanukikati へジャンプします。

それ以外の場合は、goto \*battle で最初へ戻っ  
て、バトルを続けます。

後はそれぞれが勝った時のストーリーを表示し  
ます。

これで、バトルするストーリーになりました。  
最初に入力する SARUTOBi くんの体力の値を  
いろいろ変えて、試してみてください。



バトル開始！¥

\*battle ←

;SARUTOBi くんの攻撃

\$1 の攻撃！¥

mov %2,%2-10

タヌキおやじに10のダメージ！

体力=%2¥

if %2<=0 goto \*sarutobikati ←

;タヌキおやじの攻撃

タヌキおやじの攻撃！¥

mov %1,%1-10

\$1 に10のダメージ！

体力=%1¥

if %1<=0 goto \*tanukikati ←

goto \*battle ←

;SARUTOBi くんの勝ち

\*sarutobikati

\$1 が勝ちました¥

end

;タヌキおやじの勝ち

\*tanukikati

タヌキおやじが勝ちました¥

end

「if」命令で使う条件式は、「等しい」「大きい」「小さい」など、いろいろな条件が使えます。

<code>if %1=10 goto *hitoshii</code>	;%1 が 10 と等しかったら*hitoshii へジャンプ
<code>if %1&lt;10 goto *chiisai</code>	;%1 が 10 より小さかったら*chiisai へジャンプ (10 はふくまれない)
<code>if %1&gt;10 goto *ookii</code>	;%1 が 10 より大きかったら*ookii へジャンプ (10 はふくまれない)
<code>if %1&lt;=10 goto *ika</code>	;%1 が 10 以下だったら*ika へジャンプ (10 がふくまれる)
<code>if %1&gt;=10 goto *ijou</code>	;%1 が 10 以上だったら*ijou へジャンプ (10 がふくまれる)
<code>if %1&lt;&gt;10 goto *nai</code>	;%1 が 10 と等しくなかったら*nai へジャンプ

また、後ろに書く命令は goto 以外も使えます。

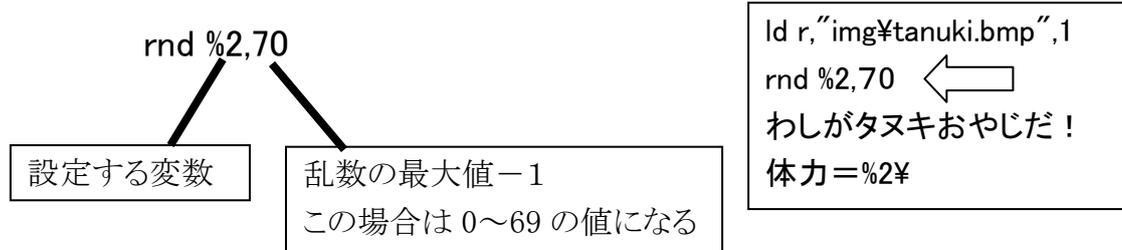
## ★乱数(らんすう)を使う

今のままだとタヌキおやじの体力が一定なので、バトルする前から勝ち負けが決まってしまうとおもしろくありません。

バトルするたびに体力の値を変えるようにしてみましょう。

そのたびに変わる数値(乱数)を使うには、`rnd`(ランダム)命令を使います。

タヌキおやじの体力%2を設定する行を、以下のようにします。



ファイルを保存して実行してみましょう。

プレイするたびにタヌキおやじの体力が変わるので、バトルに勝ったり負けたりします。

ただ、場合によっては乱数が 0 などになって、タヌキおやじがかなり弱くなってしまいます。これではおもしろくないので、体力の設定を右のようにしてみます。

まず 1 行目の `rnd` 命令で、%2 の値が 0~39 になります。

そして 2 行目の `mov` 命令で 30 が足されるので、%2 の値が 30~69 になります。

これで SARUTOBi くんという勝負になります。

```

rnd %2, 40
mov %2, %2+30
  
```

また、今は攻撃で減らされる体力が 10 で一定ですが、攻撃のたびに变化するようにしましょう。

攻撃ポイントの変数として、新しく%3 を作って、乱数で設定します。

体力(%1 または%2)を減らす所で、%3 を引き算するようにします。

これで結果がどうなるかわからなくなって、おもしろいバトルになります。

乱数の条件をいろいろ変えると、バトルの難しさが変わります。(いわゆるゲームバランス)

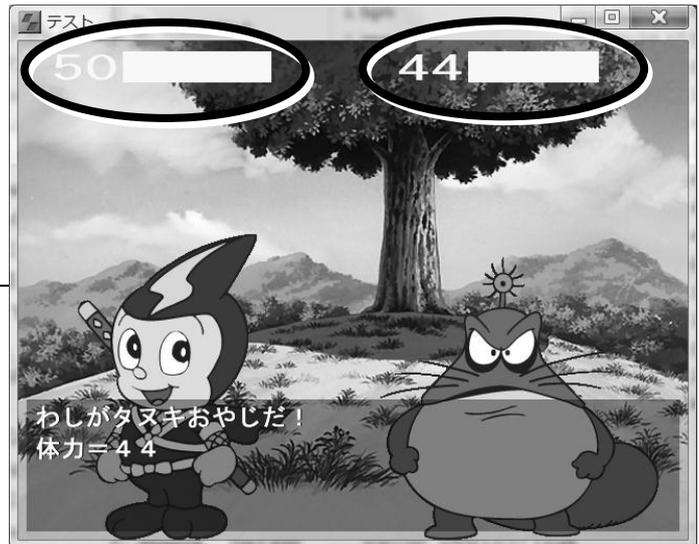
いろいろ試してみてください。

```

;SARUTOBi くんの攻撃
$1 の攻撃! ¥
rnd %3,30
mov %2,%2-%3
タヌキおやじに%3 のダメージ!
体力=%2¥
if %2<=0 goto *sarutobikati

;タヌキおやじの攻撃
タヌキおやじの攻撃! ¥
rnd %3,30
mov %1,%1-%3
$1 に%3 のダメージ!
体力=%1¥
if %1<=0 goto *tanukikati
  
```

さらに応用ですが、バトルゲームらしく、体力ゲージを画面に表示することができます。  
細かくは説明しませんが、できる人はやってみてください。



```
***バトルゲーム***
```

```
*define
caption "バトルゲーム"
deletemenu
game
```

```
*start
setwindow 16,350,25,4,24,24,0,6,20,1,1,#999999,8,345,631,471
bg "img¥haikei.jpg",1
input $1,"プレイヤーの名前は？","さるとび",20,1
input $2,"体力は？(半角文字で)","50",10,0
atoi %1,$2
```

```
;SARUTOBi くん登場
ld l,"img¥sarutobi.bmp",1
prnum 1,%1,0,10,32,32,#FFFF00
bar 1,%1,100,10,200,30,70,#FFFF00
print 1
ぼく、$1
体力=%1¥
```

SARUTOBi くんの体力ゲージを  
左上に表示

●prnum 命令: 数字(数字ラベル)を画面に表示する

```
prnum 1,%1,0,10,32,32,#FFFF00
```

数字ラベル番号,表示したい変数,x 座標,y 座標,文字の横  
サイズ,文字の縦サイズ,文字のカラーコード

```
;タヌキおやじ登場
ld r,"img¥tanuki.bmp",1
rnd %2,40
mov %2,%2+30
prnum 2,%2,330,10,32,32,#FFFF00
bar 2,%2,430,10,200,30,70,#FFFF00
print 1
わしがタヌキおやじだ！
体力=%2¥
```

●print 命令: 画面の再表示

```
print エフェクト番号
```

タヌキおやじの体力ゲージを  
右上に表示

●bar 命令: バーを画面に表示する

```
bar 1,%1,100,10,200,30,70,#FFFF00
```

バー番号,表示したい変数,x 座標,y 座標,バーの横サイズ,  
バーの縦サイズ,バー最大値,バーのカラーコード

(↓※次ページへ続く↓)

```

バトル開始！¥
*battle

;SARUTOBi さんの攻撃
$1 の攻撃！
rnd %3,30
mov %2,%2-%3
prnum 2,%2,330,10,32,32,#FFFF00
bar 2,%2,430,10,200,30,70,#FFFF00
print 1
タヌキおやじに%3 のダメージ！
体力=%2¥
if %2<=0 goto *sarutobikati

;タヌキおやじの攻撃
タヌキおやじの攻撃！¥
rnd %3,30
mov %1,%1-%3
prnum 1,%1,0,10,32,32,#FFFF00
bar 1,%1,100,10,200,30,70,#FFFF00
print 1
$1 に%3 のダメージ！
体力=%1¥
if %1<=0 goto *tanukikati

goto *battle

;SARUTOBi さんの勝ち
*sarutobikati
$1 が勝ちました¥
end

;タヌキおやじの勝ち
*tanukikati
タヌキおやじが勝ちました¥
end

```

タヌキおやじの変化した体力  
ゲージを右上に表示

SARUTOBi さんの変化した体力  
ゲージを左上に表示

※画面から数字ラベルを消したい時は `prnumclear` 命令、  
バーを消したい時は `barclear` 命令を使います。  
どちらも、直後に `print` 命令を実行してください。  
複数の数字ラベルやバーがある時は、全部消えてしまうの  
で注意してください。