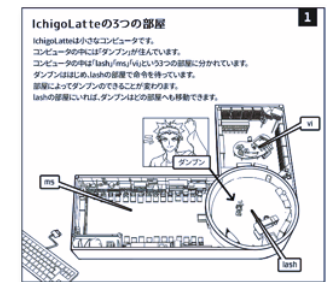


IchigoLatte でプログラミング (0)

～ LED をてんめつさせよう ～

1. `IchigoLatte 1.1.7 by na-s.jp`
`lash>`
それでは LED をあやつってみよう。
2. `lash>echo>`
まずはプログラムを真っさらに。
3. `lash>vi .`
プログラムをかく画面に、今は真っ黒。(の前のスペースに注意)
4. `led(1);`
この1行をかいたら[ESC]で戻ろう。(大文字小文字に注意)
5. `lash>ms .`
プログラムをスタート、LED ついたかな?
(*)!not found. ～"が出たりコンピュータが突然リセットしたらどこか間違い、"vi ."に戻ってなおそう
6. `lash>vi .`
では次、もう一度かく画面に。
7. `led(0);`
(*)反転部分をかき換えよう
かき変えて[ESC]。
8. `lash>ms .`
プログラムをスタート、今度は LED がきえたかな?
9. `lash>vi .`
では次、もう一度かく画面に。
10. `led(1);led(0);`
今度はつけてけすやつ、かき足したら[ESC]。

11. `lash>ms .`
スタート、ついてきえるのが見えたかな?
12. `lash>ms .`
いっしゅんだから見のがした人はもう一度。([↑]が便利)
13. `lash>vi .`
早すぎるのでなおそう。
14. `led(1);sleep(1000);led(0);`
かき足したら[ESC]。
15. `lash>ms .`
今度はしっかりついてきて1秒。
"1000"は"1,000 ミリ秒"の意味、好きな長さで光らせてみよう。
16. `lash>vi .`
では最後に 無限てんめつ をやってみよう。
17. `while(1){`
`led(1);sleep(1000);`
`led(0);sleep(1000);`
`}`
この4行、かけたら[ESC]。
18. `lash>ms .`
無限てんめつ のできあがり。([ESC]で止まるよ)
好きなてんめつリズム に改造してみよう。



プリント「<http://ichigo latte.shizentai.jp/IchigoLatte-first.pdf>」も参考にしよう!

IchigoLatte でプログラミング (1)

～無限に文字を出そう～

```
1. IchigoLatte 1.1.7 by na-s.jp  
lash>
```

それでは画面をあやつってみよう。

```
2. lash>echo>
```

まずはプログラムを真っさらに。

```
3. lash>vi .
```

プログラムをかく画面に。

```
4. log(" * " );
```

これをかいたら [ESC] で戻ろう。

```
5. lash>ms .
```

プログラムをスタート。

```
6. *lash>
```

画面に“*”が出たかな？

```
7. *lash>vi .
```

プログラムをかく画面に。

```
8. log(" * " ); log(" * " );
```

かき足して [ESC] で戻ろう。

```
9. lash>ms .
```

プログラムをスタート。

```
10. **lash>
```

こんどは2つでたかな？好きな個数を出してみよう。

```
11. lash>vi .
```

では無限に出してみよう。

```
12. while(1){  
    log(" * " );  
}
```

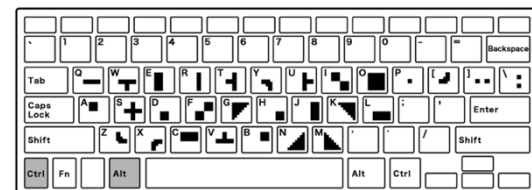
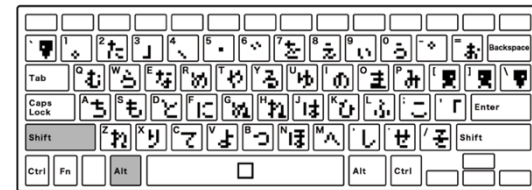
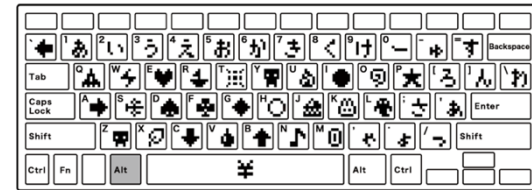
かけたら [ESC]。

```
13. lash>ms .
```

プログラムをスタートすると？

```
14. *****  
*****  
*****. . .
```

無限“*”のできあがり！（[ESC]で止まるよ）



[Alt]+[E]で♥、[Alt]+[P]で★など、[Alt]+[なにか]で絵文字もできるよ、好きな文字に改造してみよう。

NEW

log	画面に出す	プログラムを真っさらに	lash>echo>
while(1)	無限にくり返す	プログラムをかく画面	lash>vi .
		プログラムをスタート	lash>ms .

何かを無限にやりたかったら while(1){なんか}だよ



IchigoLatte でプログラミング (2)

～ 計算をしよう～

```
1. IchigoLatte 1.1.7 by na-s.jp  
lash>
```

それでは記憶をあやつってみよう。

```
2. var hiru=12;
```

“vi .” でこれを書いて “ms .” でスタート。

```
3. lash>
```

おや？ なにも起きない？

じつは起きてるよ。“var”は「記憶場所を準備せよ」、「hiru」は「記憶場所につける名前はhiru」、「=12」は「12を記憶して」、つまり「hiruという記憶場所を作って12とメモっておいて」という意味。記憶をただだからスタートしても見た目にはなにも起きない。

```
4. var hiru=12;  
log(hiru);
```

では記憶の中身を確認してみよう。“vi .” でこれをかき足す。“log”は「画面に出す」というコマンド。

```
5. 12lash>
```

プログラムをスタート、12が出た、ちゃんと記憶してる。

```
6. var hiru=12;  
log(hiru);  
hiru=11;  
log(hiru);
```

記憶を書き換えることもできるよ、“hiru=11;”は「やっぱり記憶場所 hiru は 11 ね」という意味。

```
7. 1211lash>
```

こんな感じで12と11がでたかな？

最初は12を入れてそれを画面に出したけど、途中で11に書き換えてそれも画面に出したから“1211”。

```
8. var hiru=12;  
log(hiru);log(“,”);  
hiru=11;  
log(hiru);log(“,”);
```

12と11がひっついて見にくいから間に“,”が出るように。

```
9. 12,11,lash>
```

これなら見やすい。

```
10. var hiru=12;  
log(hiru);log(“,”);  
hiru=11;  
log(hiru);log(“,”);  
var asa=6;  
log(hiru-asa);log(“,”);
```

記憶場所はいくつも作れるよ。こんどは“asa”を作って6を入れてみた。記憶場所どうして計算もできるから、“hiru-asa”で引き算して朝から昼まで何時間あるか計算もしてみたよ。

```
11. 12,11,5,lash>
```

5時間だね！

足し算は“+”、引き算は“-”、かけ算は“*”、割り算は“/”、余り算は“%”、いろんな記憶を作って計算させてみよう。

プログラムを真っさらに	lash>echo>.
プログラムをかく画面	lash>vi .
プログラムをスタート	lash>ms .

NEW

var 記憶場所を準備

IchigoLatte でプログラミング (3)

～ かずを数えよう ～

```
1. IchigoLatte 1.1.7 by na-s.jp  
lash>
```

それではかずを数えてみよう。

```
2. var c=0;  
   log(c);log(",");
```

vi でこれをかいて ms でスタート。

```
3. 0,lash>
```

そう、0 だね。”c”は記憶場所につける名前、今回はカウント(count)の‘c’を名前にしてみたよ。

```
4. var c=0;  
   log(c);log(",");  
   c=c+1;  
   log(c);log(",");
```

さあ、こんなことをやってみる。

```
5. 0,1,lash>
```

0 が 1 になった。そう、”c=c+1”は「c に 1 足したものを c に記憶する」の意味。

```
6. var c=0;  
   log(c);log(",");  
   c=c+1;  
   log(c);log(",");  
   c=c+1;  
   log(c);log(",");
```

では調子にのってこうしてみよう。

```
7. 0,1,2,lash>
```

そうだね、また 1 増えて 2。このまま 3,4,5,... と 100 までかぞえていくプログラム、つくれるかな？

… ちょっと大変だね、じゃあ無限のわざを使ってみよう。

```
8. var c=0;  
   while(1){  
     log(c);log(",");  
     c=c+1;  
   }
```

無限のわざはこう、おぼえてたかな？

```
9. 0,1,2,3,4,5,6,7,8,9,10,11,12,13,  
   14,15,16,17,18,19,20,21,22,23,24,  
   25,26,...
```

無限にかずを数えれた！ ([ESC]で止まるよ)

でも今回は 100 まで数えたい。

```
10. var c=0;  
     while(c<101){  
       log(c);log(",");  
       c=c+1;  
     }
```

そんな時はこう。”while(1)”は「無限にくり返せ」の意味だけど、”while(c<101)”は「c が 101 より小さい時はくり返せ」の意味。

(*)”while(0)”は「絶対にくり返すな」の意味

```
11. 0,1,2,3,4,5,6,7,8,9,10,11,12,13,  
    14,15,16,17,18,19,20,21,22,...,9  
    9,100,lash>
```

はい、ぴったり 100！

“c=c+2”なんてこともできるし、カウントダウン”c=c-1”もできちゃう、試してみよう！

NEW

c=c+1

c を 1 ぶやす

プログラムを真っさらに	lash>echo>.
プログラムをかく画面	lash>vi .
プログラムをスタート	lash>ms .

記憶を使えば、好きな回数だけくり返すことができるよ。

IchigoLatte でプログラミング (4)

～ カウント装置を作ろう～

```
1. IchigoLatte 1.1.7 by na-s.jp
   lash>
```

それではボタンをしらべてみよう。

```
2. log( btn() );
```

今回のプログラムはこれ、かけたらスタートしてみよう。

```
3. 0lash>
```

“0”が出る。

```
4. while(1){
   log( btn() );
}
```

無限に出してみよう。

```
5. 00000000000000000000000000000000000000000000000000000000000000000000
   00000000000000000000000000000000000000000000000000000000000000000000
   000000000000...
```

スタートすると無限に“0”が、そこで基板上的のボタンを押すと？

```
6. 00011111111111111111111111111111111111111111111111111111111111111111
   11111111111111111111111111111111111111111111111111111111111111111111
   111111111111...
```

こんどは”1”が大量に。([ESC] で止まるよ)
 “btn()”はボタンを調べるコマンド、0 か 1 を返してもらえるよ。

```
7. while(1){
   led( btn() );
}
```

“btn()”で0 か 1 がもらえるなら、こんな具合に“led()”に渡してやるとどうなる？

```
8. lash>ms .
```

プログラムをスタート。見た目には何も起きないけど基板上的のボタンを押すと LED が光る！“led(1)”はつける“led(0)”は消すだから“led(btn())”はボタンしだい！

```
9. var c=0;
   while( btn()==0 ){
     log(c);log(",");
     c=c+1;
   }
```

こんどは「カウント装置」。“while(btn()==0)”は「btn()が0 の時はくり返せ」の意味。こんな風に“btn()”はいろんな場面でつかえるよ。

```
10. lash>ms .
```

スタートすると数字がどんどんカウントアップ、基板上的のボタンを押すと止まるよ。ぴったり 1,000 で止めることできるかな？

```
11. 60,1061,1062,1063,1064,lash>
    stdin/out: uart, uart
```

ボタンで止めた時に「stdin/out: uart, uart」みたいな出てきたら uart モード(*1)になった印、あわてずもう 1 回ボタンを押そう、元に戻るよ。

ぴったり止めるのが難しすぎる？じゃあもっと簡単になるように改造してみてね！

(*1)uart モード: 他のコンピュータと通信するモード

NEW	
btn	基板のボタンをしらべる
プログラムを真っさらに	lash>echo>.
プログラムをかく画面	lash>vi .
プログラムをスタート	lash>ms .

ボタンを調べた結果の「0」「1」はいろんな場面でつかえるよ。

IchigoLatte でプログラミング (5)

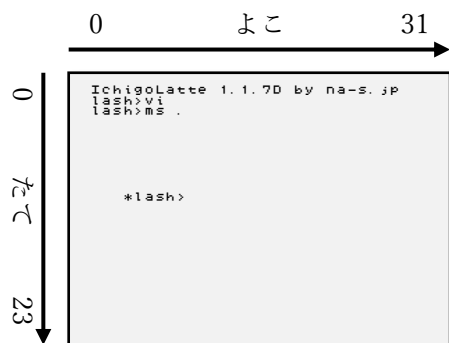
～ 弾をとばそう～

```
1. IchigoLatte 1.1.7 by na-s.jp
   lash>
```

それでは弾を飛ばしてみよう。

```
2. lc(3,10);log("*");
```

これをスタート。画面の左から3、上から10の場所に“*”が出たかな？（座標は0からはじまっているよ）



“lc(○,△)”は「場所」のコマンド。○が「よこ」△が「たて」。

```
3. cls();
   lc(3,10);log("*");
```

画面がごちゃごちゃしているから、きれいにしよう。今度はこの2行。かき足した“cls”は「画面をきれいに」というコマンド。スタートするときれいになってから“*”がでるね。

さて、この弾“*”を右に飛ばそう。「よこ」の3を4,5,6…と増やしながら描いていけば飛ぶんじゃないかな？

```
4. cls();
   lc(3,10);log("*");
   lc(4,10);log("*");
   lc(5,10);log("*");
```

というわけでかき足して4行に。スタートすると3つでた、けど速すぎて飛んだようには見えない。

```
5. cls();
   lc(3,10);log("*");sleep(500);
   lc(4,10);log("*");sleep(500);
   lc(5,10);log("*");sleep(500);
```

よし、これで3マスは飛んだふうに見えた。あとは「よこ」を6,7,8…とカウントアップしながらくり返せばいいね。

```
6. cls();
   var x=3;
   lc(x,10);log("*");sleep(500);
   x=x+1;
   lc(x,10);log("*");sleep(500);
   x=x+1;
   lc(x,10);log("*");sleep(500);
   x=x+1;
```

まず、「よこ」を“x”しちゃおう。

```
7. cls();
   var x=3;
   while(1){
     lc(x,10);log("*");sleep(500);
     x=x+1;
   }
```

そしたら無限のわざで、おっそ〜〜いビームのできあがり ^_^b。

```
8. var x=3;
   while(1){
     cls();
     lc(x,10);log("*");sleep(500);
     x=x+1;
   }
```

こんな感じに、くり返すたびに cls すれば弾っぽくなるね。速度を変えたり、弾を好きな絵文字にしてみよう！

NEW		プログラムを真つさらに	lash>echo>.
cls	画面をきれいに	プログラムをかく画面	lash>vi .
sleep	待つ	プログラムをスタート	lash>ms .

カウント装置のわざはいろんな場面で見られるよ。



IchigoLatte でプログラミング (6)

～ 英単語帳をつくろう～

```
1. IchigoLatte 1.1.7 by na-s.jp  
lash>
```

それではキーボードをつかってみよう。

```
2. While(1){  
    log( inkey() );  
}
```

今回のプログラムはこれ、かけたらスタートしてみよう。

```
3. 00000000000000000000000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000000000000000000000
```

キーボードを押していない時は“0”がいっぱい、なにかキーボードを押すと数字が出るけど速すぎて流れちゃう。こういう時は、

```
4. While(1){  
    log( inkey() );  
    sleep(1000);  
}
```

“sleep”だね。

```
5. 032000000...
```

スタートすると今度はゆっくり、いろんなキーを押してみよう。

[←]	[→]	[↑]	[↓]	[スペース]
28	29	30	31	32

出てくる数字は「キー番号」。このキー番号をみれば何が押されたかわかるってしかけ。1回押したら1回だけ取れて、2回押したら2回とれるね。

```
6. While(1){  
    if(inkey()==32) log("Hi!");  
}
```

はい、あいさつ装置のできあがり。

```
7. Hi!
```

スタートしたらキーを押そう、[スペース]の時だけあいさつしてくれるよ。“if(inkey()==32)”は「もしキー番号が32なら」の意味。[↑]キーを押したらあいさつするように改造できるかな？

```
8. While(1){  
    var k=inkey();  
    if(k==97) log("Apple,");  
    if(k==98) log("Bubble,");  
    if(k==99) log("Cable,");  
}
```

お次は英単語帳のできあがり。スタートしたら[a]や[b]や[c]のキーを押してみよう。“k=inkey()”は「キーボードからのキー番号をkという記憶場所にメモ」という意味。“if(k==97)”は「もしkにメモしたキー番号が97なら」の意味。“97”は[a]キーだよ。

a	97	g	103	m	109	s	115	y	121
b	98	h	104	n	110	t	116	z	122
c	99	i	105	o	111	u	117	A	65
d	100	j	106	p	112	v	118	B	66
e	101	k	107	q	113	w	119	C	67
f	102	l	108	r	114	x	120	:	:

小文字の[a]から[z]までの番号はこんな感じ。大文字の[A]は65からはじまってるよ。このキー番号表は「ASCII(アスキー)コード表」といって世界共通の番号表だよ！

好きなアルファベットの単語も加えてみよう！

NEW		
inkey	キー番号をゲット	プログラムを真っさらに lash>echo>.
if	もし	プログラムをかく画面 lash>vi .
		プログラムをスタート lash>ms .

if() をつかえば特定の時だけ何かをやるよ。



IchigoLatte でプログラミング (7)

～ 画面の中を散歩しよう ～

```
1. IchigoLatte 1.1.7 by na-s.jp
   lash>
```

それでは主人公を表示してみよう。

```
2. cls();
   lc(16,12);log("a");
```

スタートすると画面の真ん中らへんに主人公がでるね。画面はよこ0～31、たて0～23なので、(16,12)はだいたい真ん中。

```
3. while(1){
   cls();
   lc(16,12);log("a");
}
```

散歩しているように見せるのにパラパラマンガのわざを使おう。こんな風に「消し(`cls`)て描く(`lc`, `log`)」を無限にくり返しておくとか超高速に主人公が同じ場所に描かれるよ、スタートしてみよう。あとは「よこ」に動くために「16」を17,18,19,...と変えていけばアニメーションしていくかも!?

```
4. var x=16;
   while(1){
   cls();
   lc(x,12);log("a");
}
```

さあ、「よこ」に動くために「16」を“`x`”にメモ、“`lc`”に“`x`”を指定したよ。これをスタートしても見た目はさっきと同じだけど、“`x`”の数を変更できればよこ方向にうごける、って計画。

```
5. var x=16;
   while(1){
   cls();
   lc(x,12);log("a");
   if(inkey()==29) x=x+1;
}
```

さあ、「消す」「描く」を無限にくり返している間に、キー番号をチェックしておこう。“`if(inkey()==29)`”、つまり「もしキー番号が29」だったら“`x=x+1`”、つまり「`x`を1つ増やす”。“29”は[→]キーだよ、散歩できたかな?

```
6. var x=16;
   while(1){
   cls();
   lc(x,12);log("a");
   var k=inkey();
   if(k==28) x=x-1;
   if(k==29) x=x+1;
}
```

[←]キーは“28”、左に行くには“`x`”を1つ減らすよ。“`inkey`”は「キーボードを1回押したら1回だけ取れる」のおぼえてるかな?だから“`inkey`”で取るのは1回だけにしてそれを“`k`”にメモして、“`k`”が“28”か“29”かを調べることにしたよ。

```
7. var x=16;
   var y=12;
   while(1){
   cls();
   lc(x,y);log("a");
   var k=inkey();
   if(k==28) x=x-1;
   if(k==29) x=x+1;
   if(k==30) y=y-1;
   if(k==31) y=y+1;
}
```

さあたてにも動こう、「12」を“`y`”にしたよ。これで散歩する主人公のできあがり!

主人公は絵文字に変更しようだい!

プログラムを真っさらに	lash>echo>.
プログラムをかく画面	lash>vi .
プログラムをスタート	lash>ms .

NEW

x=x-1	xを1へらす
-------	--------

アニメーションの原理は「パラパラマンガ」だよ。



IchigoLatte でプログラミング (8)

～ 地球を散歩しよう ～

```
1. IchigoLatte 1.1.7 by na-s.jp
   lash>
```

それでは地球を散歩してみよう。

```
2. var x=16;
   var y=12;
   while(1){
     cls();
     lc(x,y);log(" @");
     var k=inkey();
     if(k==28) x=x-1;
     if(k==29) x=x+1;
     if(k==30) y=y-1;
     if(k==31) y=y+1;
   }
```

これは前回のプログラム。スタートして画面外に散歩するといなくなっちゃうの、ちょっとさみしい。

```
3. ~~~~~
   if(k==30) y=y-1;
   if(k==31) y=y+1;
   //TikyU
   if(31<x) x=0;
}
```

(*)~~~~~...~は「プログラムの途中部分」の意味だよ

画面のよこは0~31、だから「もし \times が 31 より大きく」になったら " $\times=0$ " にしてしまおう。プログラムの下の方にこの2行を追加、スタートして右はじをはみ出ると、なんと左から登場！//TikyU は「コメント」といって、プログラムを見やすくするためのもの。

```
4. ~~~~~
   //TikyU
   if(31<x) x=0;
   if(x<0) x=31;
}
```

さっきの逆、左に消える時に右から出るにはこの1行だね、スタートしてみよう。右に進めば左から現れ、左に進めば右から、まるで丸い地球を散歩してるみたい！

```
5. ~~~~~
   //TikyU
   if(31<x) x=0;
   if(x<0) x=31;
   if(23<y) y=0;
   if(y<0) y=23;
}
```

さあ、"y" もくわえて上と下もつながるようにしてみたよ！"if" 4行のおかげで、画面の中どこへ行ってもはみ出ることがなくなったね！

(*)上に行くとか下からでてくるなんて、地球というよりはドーナツだね！

```
6. ~~~~~
   //TikyU
   x=x%32;
   y=y%24;
}
```

じつは "if" を4つ使わなくても、これで右から左、下から上につながるよ！"%" は「割り算のあまり」の意味、 $31 \div 32$ のあまりは 31、 $32 \div 32$ のあまりは 0 でしょ？だから右と左がつながるんだ！

```
7. ~~~~~
   //TikyU
   x=(x+32)%32;
   y=(y+24)%24;
}
```

左から右、上から下の場合は割り算がマイナスになるからうまくいかない。だからこうやってマイナスにならないようにしてやるといいね！

同じことをやるにもやり方はいくつもあるよ！どれが正解とかでなく、自分で思いついた方法がとってもかっこいいよ！

NEW		
	プログラムを真っさらに	lash>echo>.
//***	コメント	lash>vi .
x=x%32	x を 32 で割った余り	lash>ms .

割り算のあまりは回るんだよ。



IchigoLatte でプログラミング (9)

～ ドラゴンキューブをゲットしよう～

```
1. IchigoLatte 1.1.7 by na-s.jp
   lash>
```

それではドラゴンキューブをだしてみよう。

```
2. var x=16;
   var y=12;
   while(1){
     cls();
     lc(x,y);log("a");
     var k=inkey();
     if(k==28) x=x-1;
     if(k==29) x=x+1;
     if(k==30) y=y-1;
     if(k==31) y=y+1;
     //Tikyuu
     x=(x+32)%32;
     y=(y+24)%24;
   }
```

これは前回のプログラム。

```
3. ~~~~~
   cls();
   lc(6,9);log("1");
   lc(x,y);log("a");
   ~~~~~
```

この1行をついか、イーシンキューブがあらわれるよ。

```
4. ~~~~~
   var y=12;
   var dx=rnd(32),dy=rnd(24);
   while(1){
     ~~~~~
     lc(dx,dy);log("1");
     ~~~~~
```

ドラゴンキューブはどこにあらわれるかわからない、神のキューブ。こうすればスタートするごとにランダムな場所にあらわれるよ。

```
5. ~~~~~
   lc(x,y);log("a");
   if((x==dx)*(y==dy)) break;
   var k=inkey();
   ~~~~~
```

主人公がドラゴンキューブにふれる時、 x と dx が同じ、かつ、 y と dy が同じ、になるよね？“($x==dx$)*($y==dy$)”の“*”は「かつ」(*1)という意味。“break”は「直ちに“while”をやめ」の意味。スタートしてドラゴンキューブにふれると“while”をぬけて、プログラムが終了するね。

(*1)「または」は“+”

```
6. var x=16;
   var y=12;
   while(1){
     var dx=rnd(32),dy=rnd(24);
     while(1){
       cls();
       lc(dx,dy);log("1");
       ~~~~~
     }
     y=(y+24)%24;
   }
```

“break”で“while”をぬけたら、次のアルシクキューを出すためにまた“rnd”のところに行きたい。だから外側をもう1個“while”でくるんだよ(*2)。スタートしてドラゴンキューブをゲットしてみよう、次のキューブがあらわれるね！

(*2)くるまれた内側の行には先頭にスペースを1個ずつたして見やすくしたよん？イーシンキューブ“1”の次がアルシクキューブ“2”じゃないって？カウント装置のわざ、おぼえてるでしょ！

NEW

rnd(32)	32より小さいランダムな数
if(~*~)	かつ
if(~+~)	または
break	直ちに while をやめ

プログラムを真っさらに	lash>echo>.
プログラムをかく画面	lash>vi .
プログラムをスタート	lash>ms .

if()には「*」かつ」と「+」または」が使えるよ。

IchigoLatte でプログラミング (a)

～ シンリユウを呼びだそう～

```
1. IchigoLatte 1.1.7 by na-s.jp  
lash>
```

それではドラゴンキューブを7つ集めよう。

```
2. var x=16;  
var y=12;  
var d=1;  
while(1){  
var dx=rnd(32), dy=rnd(24);  
while(1){  
cls();  
lc(dx, dy); log(d);  
lc(x, y); log("2");  
if((x==dx)*(y==dy)) break;  
var k=inkey();  
if(k==28) x=x-1;  
if(k==29) x=x+1;  
if(k==30) y=y-1;  
if(k==31) y=y+1;  
//Tikyuu  
x=(x+32)%32;  
y=(y+24)%24;  
}  
d=d+1;  
}
```

前回のプログラムに3箇所つか、「カウント装置」のわざだね。スタートすると、イーシンキュー“1”、アルシンキュー“2”、サンシンキュー“3”、…と無限にゲットできるね!

```
3. var x=16;  
var y=12;  
var d=1;  
while(d<8){  
var dx=rnd(32), dy=rnd(24);  
while(1){  
~~~~~
```

無限じゃだめ、ドラゴンキューブは全部で7つなんだ。だから“while(d<8)”。さあ、これで7つゲットしたらプログラムが終了するようになったね!

```
4. ~~~~~  
y=(y+24)%24;  
}  
d=d+1;  
}  
  
//sinryuu!!!  
cls();  
lc(10, 10); log("おが°いはなんだ°?");  
lc(10, 11); log("=====-");
```

さあ、最後に神の龍「シンリユウ」の登場だ、7つ集めてみよう!

```
5. var x=16;  
var y=12;  
var d=1;  
var t=tick();  
while(d<8){  
~~~~~  
//sinryuu!!!  
cls();  
if(tick()-t < 20000){  
lc(10, 10); log("おが°いはなんだ°?");  
lc(10, 11); log("=====-");  
}  
else{  
lc(10, 11); log("...");  
}  
}
```

いつもシンリユウがでるから、素早くあつめた時だけ出るようにしよう。“tick”はストップウォッチ、コンピュータの裏方でいつも動いているよ。スタートした時に **t** にメモっておいて、シンリユウがでる時の“tick”と引き算しよう。20,000 ミリ秒より早かった時だけシンリユウがでるよ、“else”は「そうじゃなかったら」の意味。ドラゴンキューブ、完成かな?

NEW

tick	ストップウォッチ
else	ifじゃなかったら

プログラムを真っさらに	lash>echo>.
プログラムをかく画面	lash>vi .
プログラムをスタート	lash>ms .