

画面制御エスケープシーケンス

yamotonalds

2017-06-19

画面制御エスケープシーケンス

pecoっぽいツールはどうやったら作れるか？

そもそも画面消して何か表示してそれ消して元に戻ってくるってどうやるの？

→エスケープシーケンスでできるっぽい

エスケープシーケンスとは

文字以外の何かを制御するための特殊な数値の連続で、エスケープから始まるやつ

エスケープって？

	00	10	20	30	40	50	60	70
00	NUL	DLE	SP	0	@	P	`	p
01	SOH	DC1	!	1	A	Q	a	q
02	STX	DC2	"	2	B	R	b	r
03	ETX	DC3	#	3	C	S	c	s
04	EOT	DC4	\$	4	D	T	d	t
05	ENQ	NAK	%	5	E	U	e	u
06	ACK	SYN	&	6	F	V	f	v
07	BEL	ETB	'	7	G	W	g	w
08	BS	CAN	(8	H	X	h	x
09	HT	EM)	9	I	Y	i	y
0A	LF	SUB	*	:	J	Z	j	z
0B	VT	ESC	+	;	K	[k	{
0C	FF	FS	,	<	L	\	l	
0D	CR	GS	-	=	M]	m	}
0E	SO	RS	.	>	N	^	n	~
0F	SI	US	/	?	O	_	o	DEL

ESCの表記

- 0x1b (16進数)
- 033 (8進数)
 - 27 (10進数)
- \e
- ^[

エスケープシーケンスでできる こと

- カーソル移動・テキストクリア
 - CSI (Control Sequence Introducer)
- 色・スタイル変更
 - SGR (Select Graphics Rendition)
- 表示倍率変更

等

例

`\e[2J` 画面クリア

`\e[n;mH` n行m列目にカーソルを移動

`\e[?25l` カーソルを非表示にする

`\e[?25h` カーソルを表示する

`\e[31m` 文字を赤にする

コーディング例

シェル

```
print "\x1b[38;5;208mりんご\x1b[m"  
print "\033[38;5;208mりんご\033[m"  
print "\e[38;5;208mりんご\e[m"
```

Ruby

```
print "\x1b[38;5;208mりんご\x1b[m"  
print "\033[38;5;208mりんご\033[m"  
print "\e[38;5;208mりんご\e[m"
```

Go

```
fmt.Print("\x1b[38;5;208mりんご\x1b[m")  
fmt.Print("\033[38;5;208mりんご\033[m")
```

实例

くるくる回るやつ

```
for i in `seq 1 10`  
do  
    printf "\e[D|" ; sleep 0.1;  
    printf "\e[D/" ; sleep 0.1;  
    printf "\e[D-" ; sleep 0.1;  
    printf "\e[D\\" ; sleep 0.1;  
done
```

が降るスクリプト

何年か前に話題になった

```
$ ruby -e 'C=`stty size`.scan(/\d+/)[1].to_i;S="\xf0\x9f\x8d\xa3";a={};puts "\033[2J";loop{a[rand(C)]=0;a.each{|x,o|a[x]+=1;print "\033[#  
{o};#{x}H \033[#{a[x]};#{x}H#{S} \033[0;0H"};$stdout.flush;sleep 0.01}'
```


少し読みやすくすると

```
C=`stty size`.scan(/\d+/)[1].to_i
S="  "
pos={}
puts "\033[2J"
loop {
  pos[rand(C)]=0
  pos.each { |col,old_row|
    pos[col]+=1
    print "\033[#{old_row};#{col}H \033[#{pos[col]}#
  }
  $stdout.flush
  sleep 0.01
```

自分でも動かしてみた

おわりに

- エスケープシーケンスで文字の色変えたり動かしたりできる
- 入力のこととも考えるとツール作るときはライブラリ使った方が良さそう

iTerm2での画像表示

ESC] 1337 ; File = [optional arguments] : base-64 encoded file contents ^G

```
filename = 'serval_jump.png'  
data = `base64 < #{filename}`  
char = "\033]1337;File=inline=1:#{data}\a"
```