

シェルスクリプトで 頭の体操

第1回いたこ会 @ USP研究所

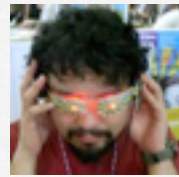
2010-08-28

たかのみつひろ / @takano32 <tak@no32.tk>

グリー株式会社 / 日本UNIXユーザ会



自己紹介



- @takan032 / TAKANO Mitsuhíro
- Japan UNIX Society
- Ruby committer (IA-64 maintainer)
- the Earth and the Internet

最近の活動

- JANOG 26 / InternetWeek 2010
- LLTV / LL Tiger
- 破滅*ラウンジ in 台湾
- 転職
 - WEBの基盤



シェルスクリプト

- お手軽
- トリッキーに使える
- パズル的な動き
- 便利だけど楽しく遊ぶこともできる

テーマ

- 「シェルスクリプト」
- @yusukebe もくる
 - 「エロ」
- 基本のテーマは「頭の体操」

頭の体操 + 工口 = ...

Brainf*ck

Brainfuckとは

- 簡単なルールのみ言語
 - 「難解プログラミング言語」
- 言語の学習のために使われる
- チューリング完全な言語
 - 理論上はC言語などと同じ能力

Brainf*ckの基本

- ポインタの操作
- 値の操作
- 入出力
- 反復制御

ポインタの操作

- “>”

- ptr++;

- “<”

- ptr--;

値の操作

□ “+”

□ $(*ptr)++;$

□ $*ptr = *ptr + 1;$

□ “-”

□ $(*ptr)--;$

□ $*ptr = *ptr - 1;$

入出力

□ “.”

□ `putchar(*ptr);`

□ “,”

□ `*ptr = getchar();`

反復制御

□ “[

□ while(*ptr) {

□ “[

□ }

よくわからん

Hello World

```
+++++++[[>+++++++>+++++++>+++++
+<<<-]>.>+ +.+++++. .+++.>- .
-----.<+++++.----- .++
+ .----- .-----.>+ .
```


やっぱりよくわからん

千里の道も一歩から

e

注目

+++++[>++++>++++>++++
+<<<-]>.>+ +.+++++. + +.>-.
-----.<+++++.-----+. +
+.-.-----.>+.

H

L

O

“llo”

+++++[>++++>++++>++++
+<<<-]>.>+ +.+++++. .+++.>-.
-----.<+++++.-----+. +
+.------.->+.

l

o

千里の道も一歩から

はじめから

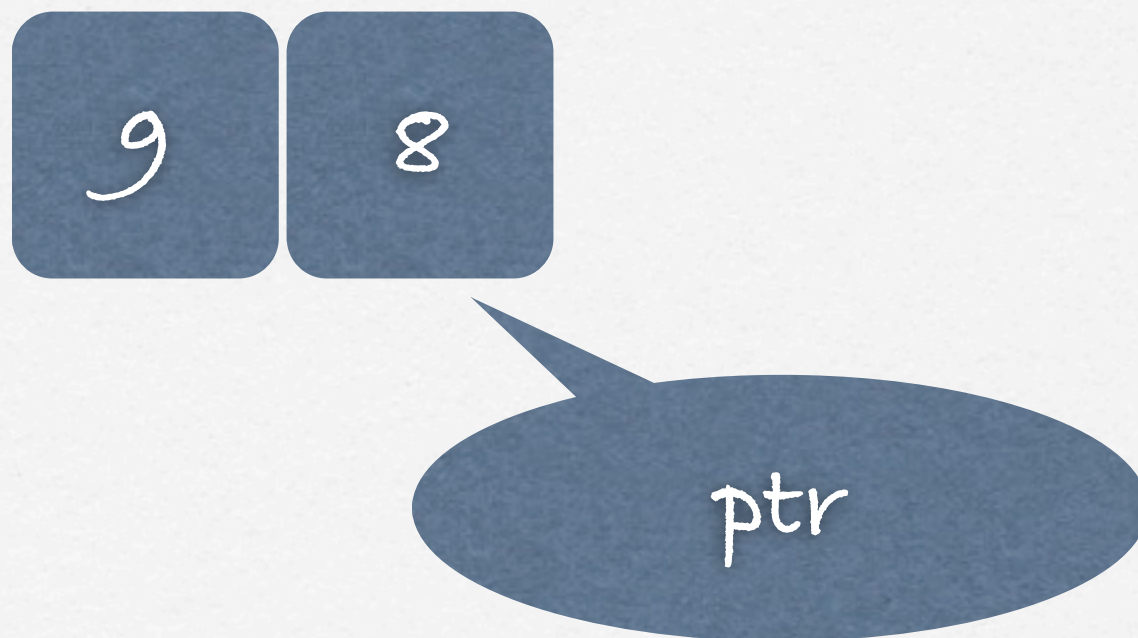
```
+++++[>++++>++++>++++  
+<<<-]>.>+ +.+++++. + +.>-.  
-----.<+++++.-----+. +  
+.------.->+.
```

9

ptr

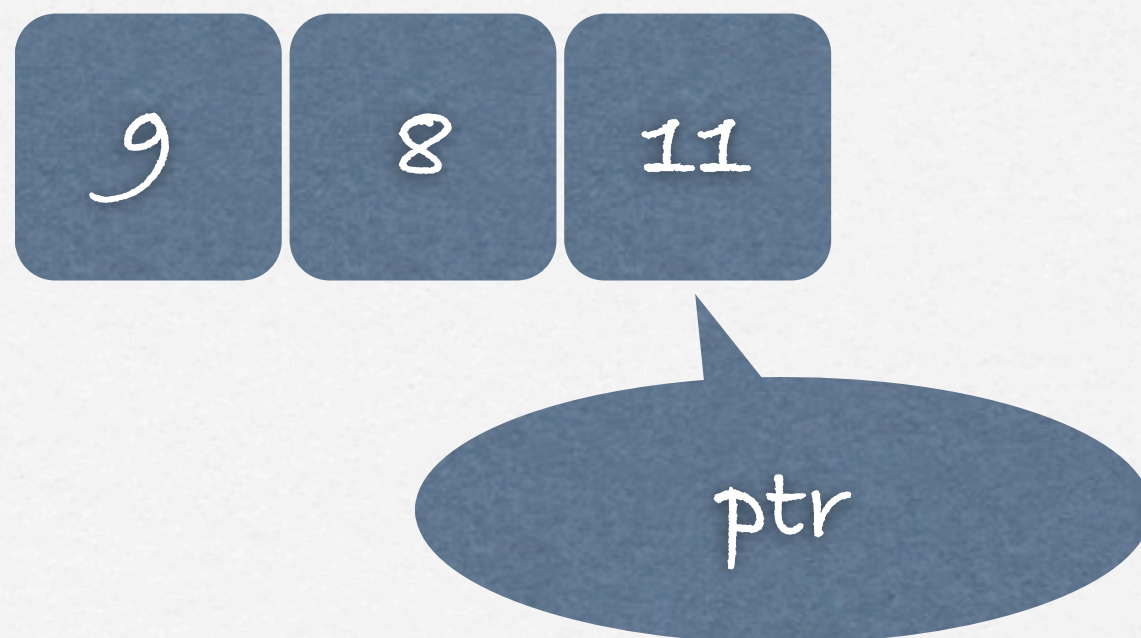
8

```
+++++[>++++++>++++>++++
+<<<-]>.>+ +.+++++. .+++.>-.
-----.<+++++.-----+.
+.------.->+.
```



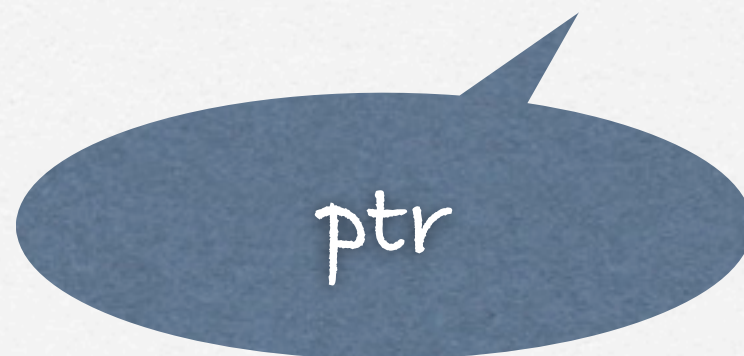
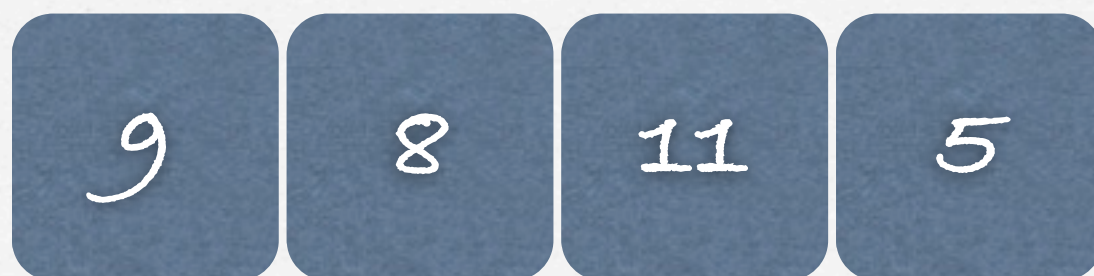
11

```
+++++[>++++++>++++++>+++++
+<<<-]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



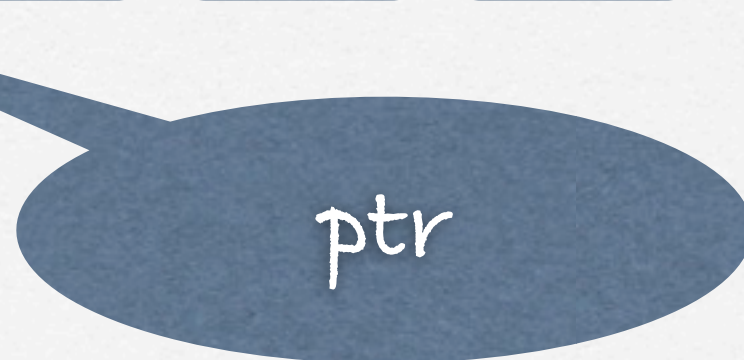
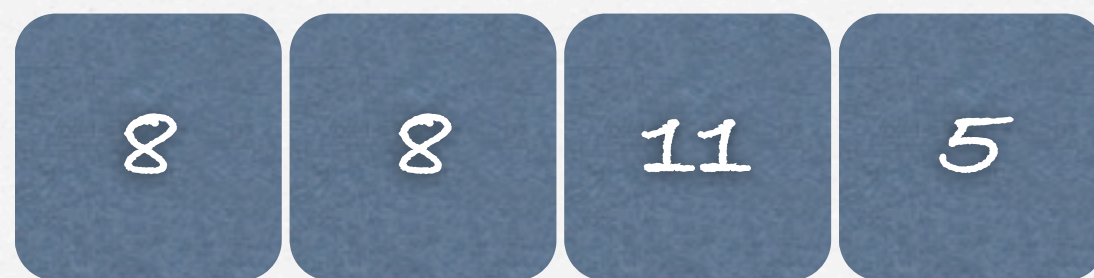
5

```
+++++[>++++++>++++++>+++++
+<<<-]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



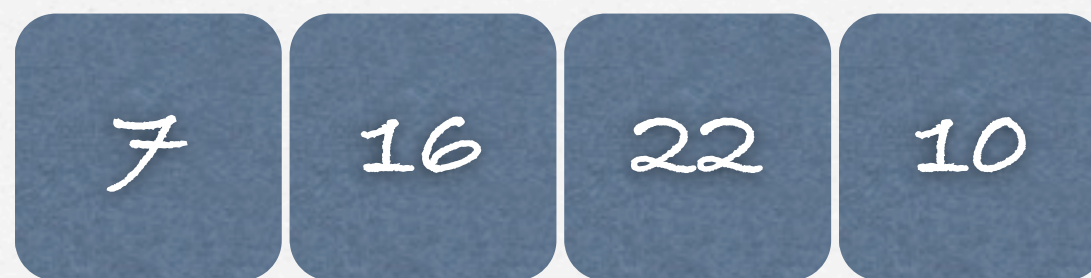
9 → 8

```
+++++++[>++++++>++++++>+++++
+<<<-]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



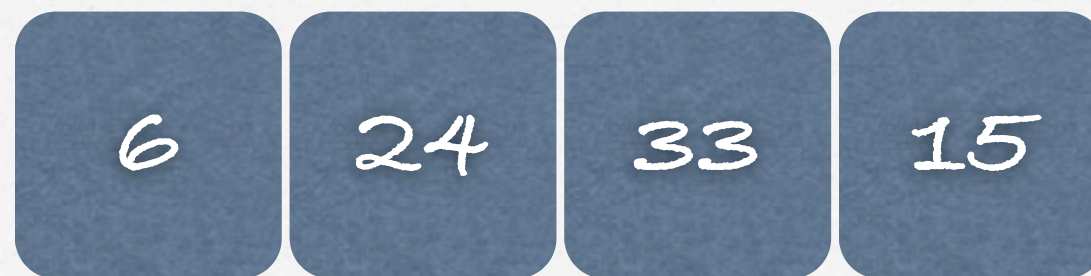
繰り返し

```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



繰り返し

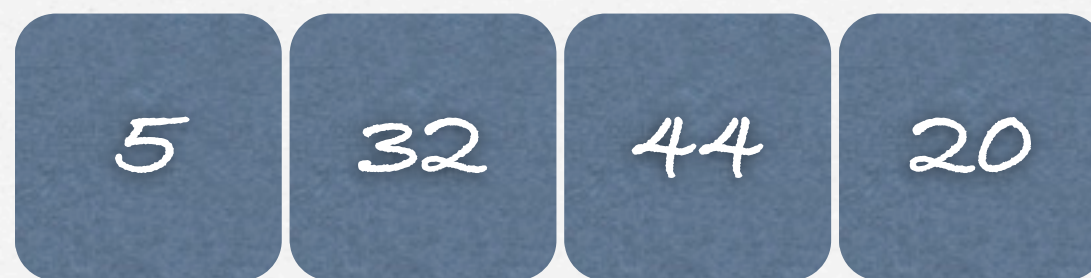
```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



ptr

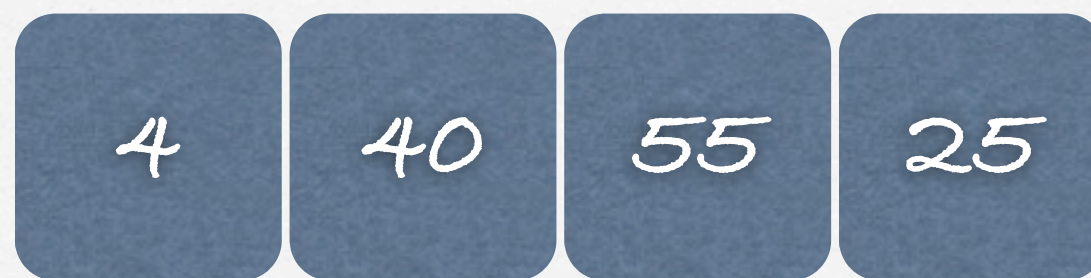
繰り返し

```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ] > . > ++ . +++++ . . + + + . > - .
----- . <+++++++ . ----- . ++
+ . ----- . ----- . > + .
```



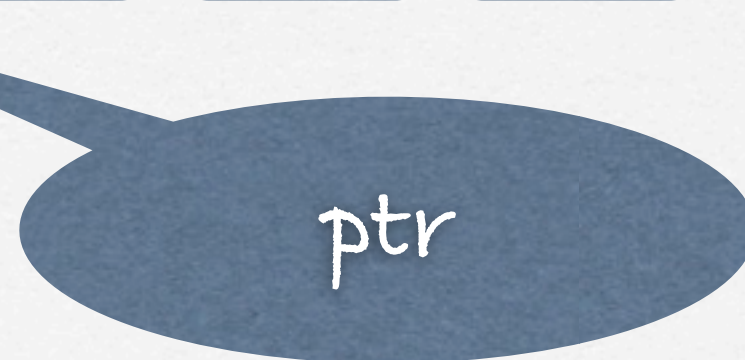
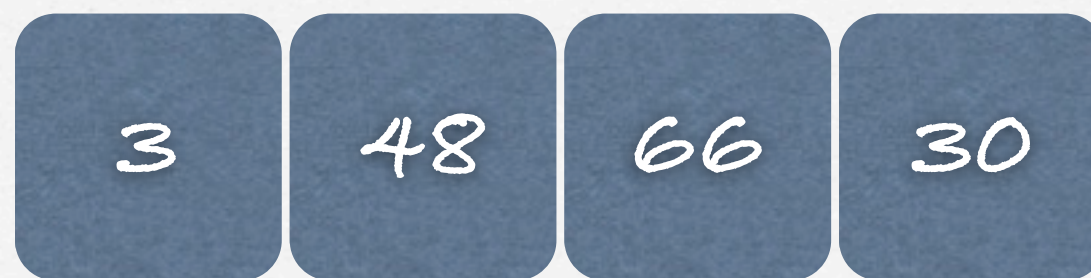
繰り返し

```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



繰り返し

```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ]>.>++.+++++. .+++.>-.
-----.<+++++.-----++.
+.------.->+.
```



繰り返し

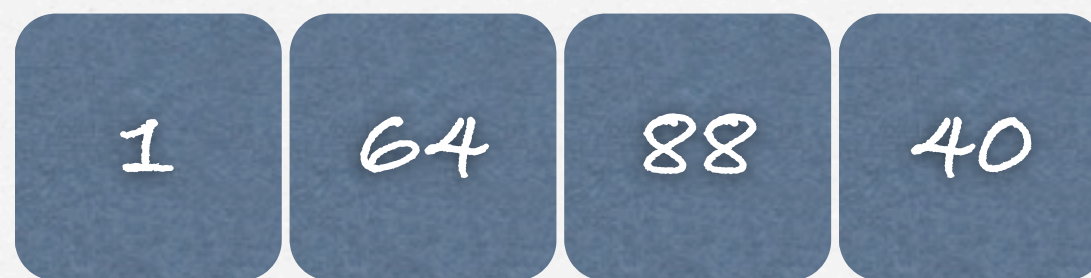
```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



ptr

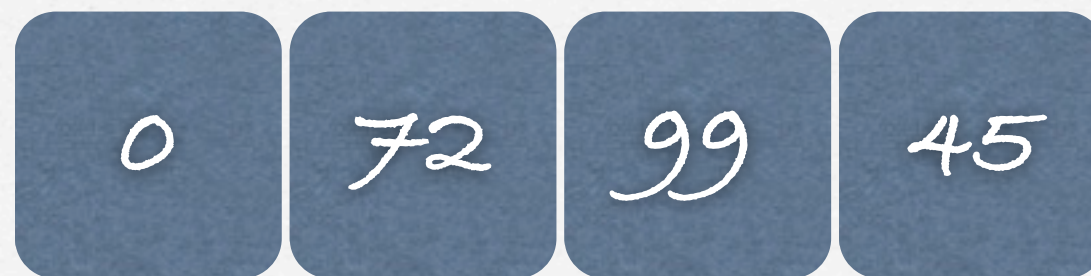
繰り返し

```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ]>.>++.+++++.+.+.>-.
-----.<+++++.-----++.
+.------.->+.
```



繰り返し

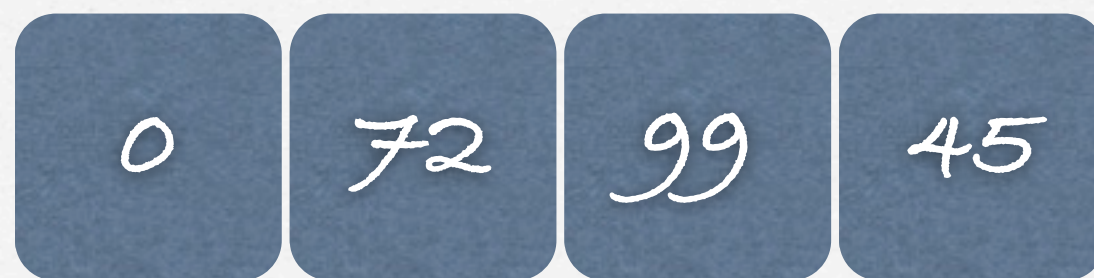
```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ] > . > ++ . +++++ . . + + . > - .
----- . <+++++++ . ----- . ++
+ . ----- . ----- . > + .
```



ptr

“H”

```
+++++++ [ >+++++++>+++++++>+++++
+<<<- ]> .>++. .+++++. .+++>-.
-----<+++++.----->+.
+ .----->+.
```



“e”

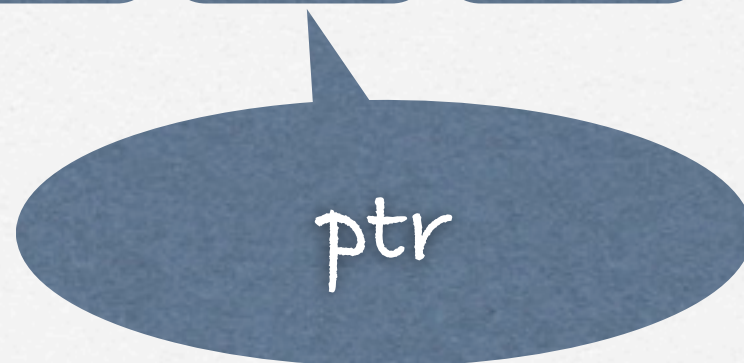
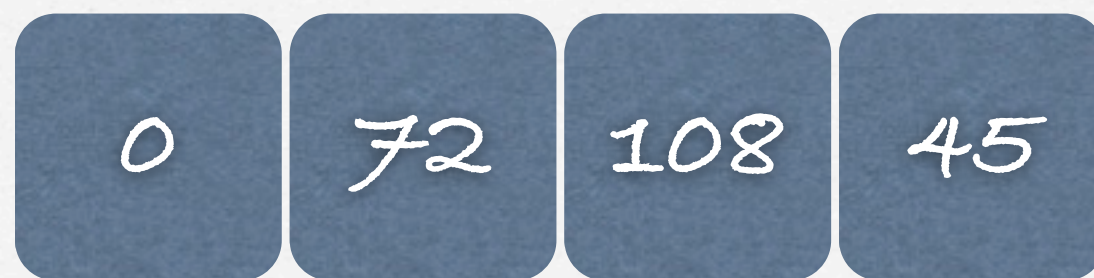
```
+++++[>++++++>++++++>++++
+<<<-]>.>++.+++++.+++.>-.
-----.<+++++.-----++.
+.------.->+.
```

0	72	101	45
---	----	-----	----

ptr

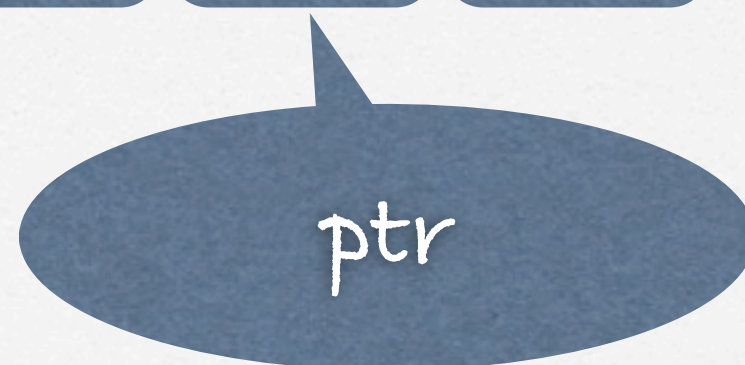
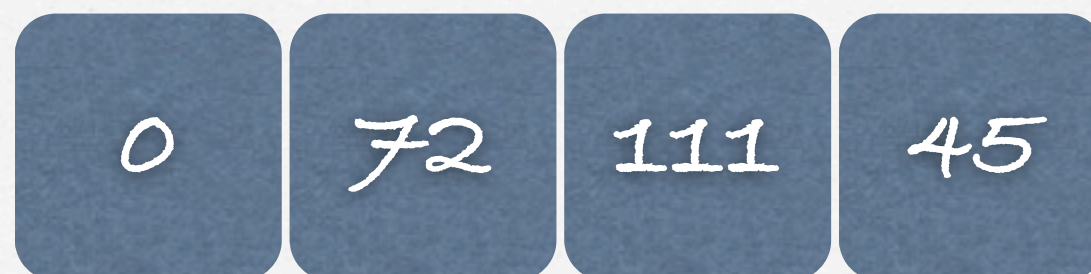
“|”, “|”

```
+++++++[>+++++++>+++++++>+++++
+<<<-]>.>++.+++++.+.+++.>-.
-----.<+++++++.------++.
+.------.------.>+.
```



“0”

```
+++++[>++++++>++++++>++++
+<<<-]>.>++.+++++.+++.>-.
-----.<+++++.-----++.
+.------.->+.
```



“ ”
，

```
+++++++[>++++++>++++++>+++++
+<<<-]>.>++.+++++. .+++.>- .
-----.<+++++.----- .++
+.------.->+.
```

0	72	111	44
---	----	-----	----

ptr

やっと“Hello,”

続きは**Web**で

本題

- Brainf*ck処理系を作ったよ
- シェルスクリプトで実装
- Brainf*ck \rightarrow s/^ (.*)ck\$/¥1sh/

Brainfush

Brainfush

- 伏せ字にする必要がない！！！！
- 手を抜いてBrainf*ckを実装
- シェルをフル活用
 - 一度の実行で二度使う

実装方針

- 入力の読み取り

 - ライン → 文字 → 解析

- トランスレータ

 - Brainf*ckコードの変換

実装

□ ">"

□ $PTR = \$((\$PTR + 1))$

□ "+"

□ $VAL = VAL_PTR$

$VAL = \$((\$VAL))$

$eval\ VAL_PTR = \$((\$VAL + 1))$

実装

□ “.”

□ `VAL=VAL_${PTR}`

`VAL=$((${VAL}))`

`OCH=`echo "obase=8; ${VAL}" | bc``

`OCH=`echo "¥0${OCH}"``

`/bin/echo -n "${OCH}"`

コラム

- ASCIIコードで文字出力

- `echo "¥0xxx"`

- `xxx` は八進数

- `echo -n hoge`

- 改行なしの出力

実装

□ "[

□ VAL=VAL_\$PTR

VAL=\$((\$VAL))

while ["VAL" != "0"]; do

□ "]"

□ VAL=VAL_\$PTR

VAL=\$((\$VAL)); end

実行

- Brainfuckコードの変換
 - シェルスクリプトに変換
- 実行
 - シェルスクリプトに喰わせる

デモ

- `/bin/sh bf2sh.sh < hello.bf`
- Brainf*ckコード→シェルスクリプト
- `/bin/sh bf2sh.sh < hello.bf | /bin/sh`
- Brainf*ck コードの実行

まとめ

- シェルはインタプリタ
- 動的に評価することができる
- シェルに評価させたいスクリプトを生成できれば大勝利！！！！
- ほかの言語処理系も実装できる？

Brainfush

- `github.com`

- `http://g.c./takan032/brainfush`

- `slideshare`

- `http://w.s.n./takan032/brainfush`

ご清聴ありがとうございました

