

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



- @takano32 / TAKANO Mitsuhiro
- 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

Brainf*ckとは

- 簡単なルールのみの言語
- 「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();

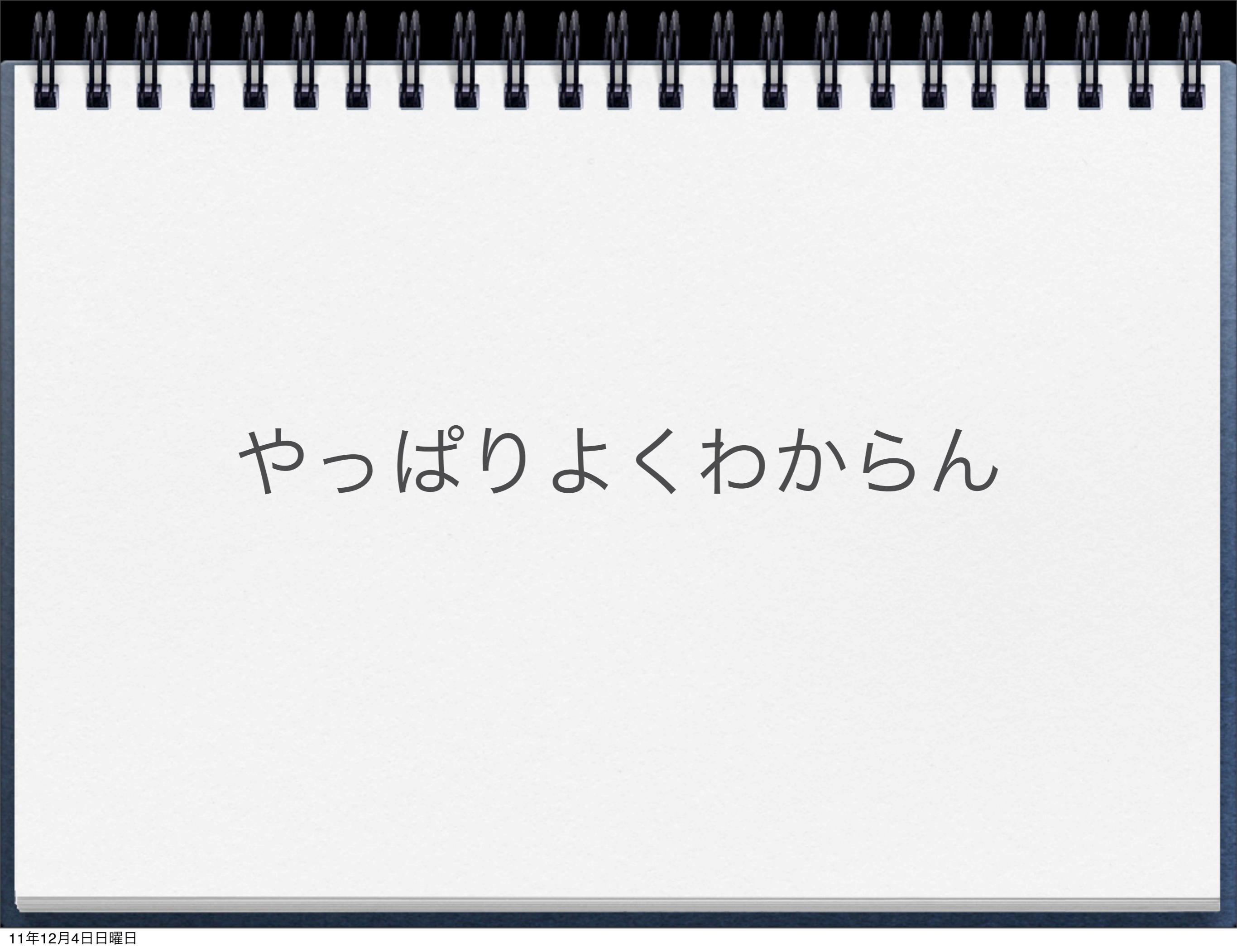
反復制御

- “[”
- while(*ptr) {
- “]”
- }

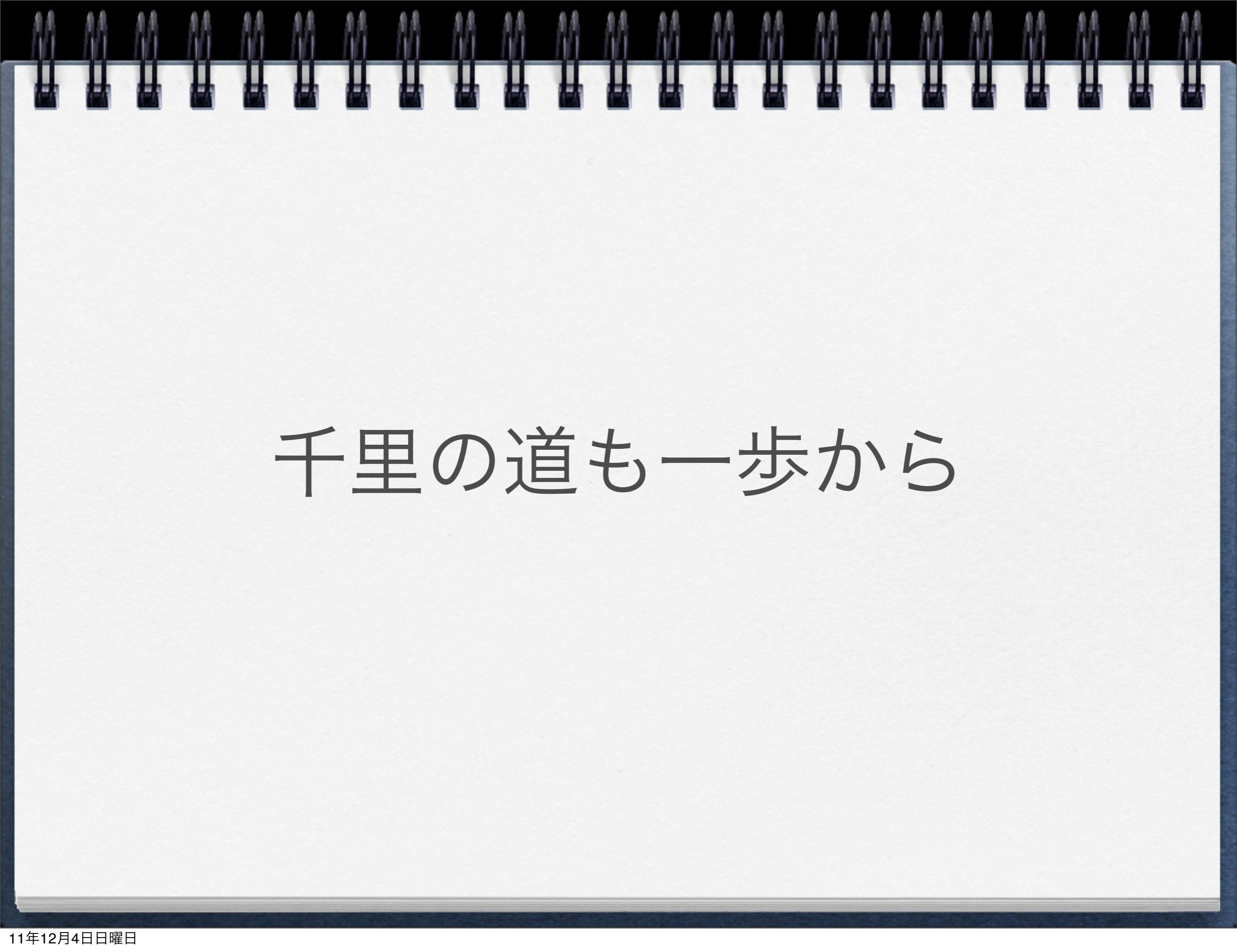
よくわからん

Hello World

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

This image shows a single, blank page from a spiral-bound notebook. The page is white with light blue horizontal ruling. It features a dark blue spiral binding along the top edge. The right side of the page shows the dark blue cover of the notebook.

やっぱりよくわからん

This image shows a single, blank page from a spiral-bound notebook. The page is white with light blue horizontal ruling. It features a dark blue spiral binding along the top edge. The paper has a slightly textured appearance.

千里の道も一歩から

注目

e

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

H

L

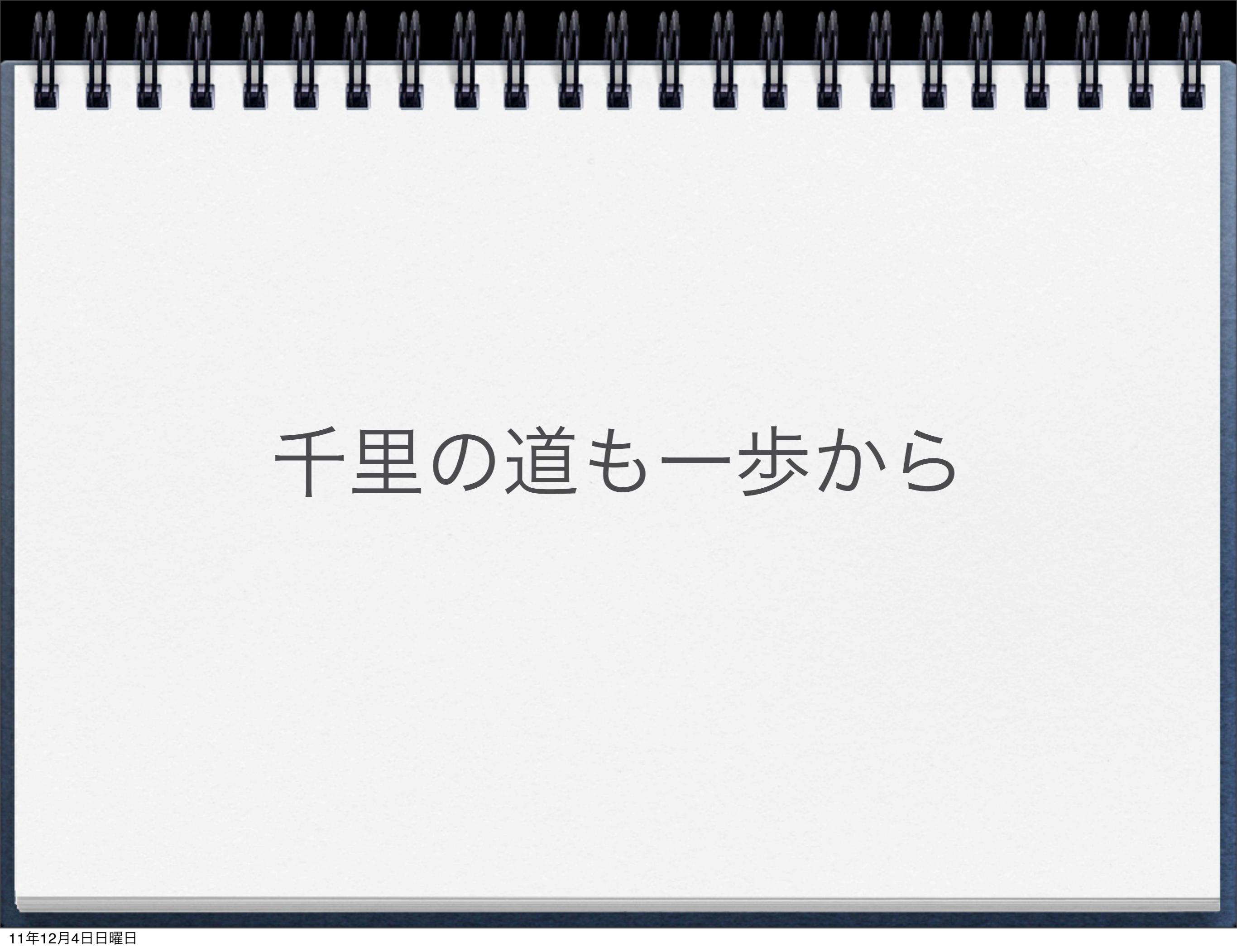
O

“||o”

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

l

o

This image shows a single, blank page from a spiral-bound notebook. The page is white with light blue horizontal ruling. It features a dark blue spiral binding along the top edge. The paper has a slightly textured appearance.

千里の道も一歩から

はじめから

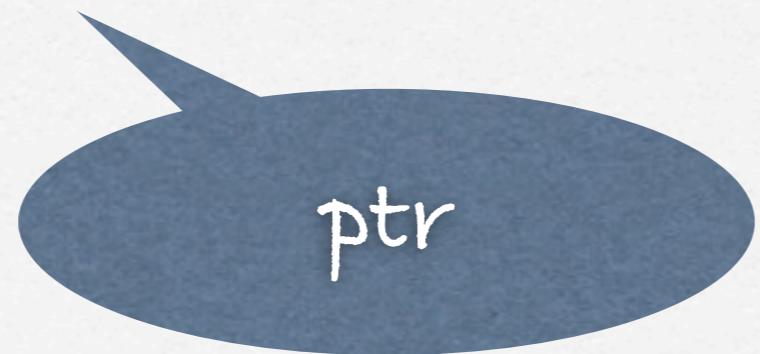
```
++++++[>+++++>++++++>++++  
+<<<-]>. >++. ++++++. .+++. >-.  
-----.<+++++++. -----. ++  
+. -----.<-----.>+.
```

9

ptr

8

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



11

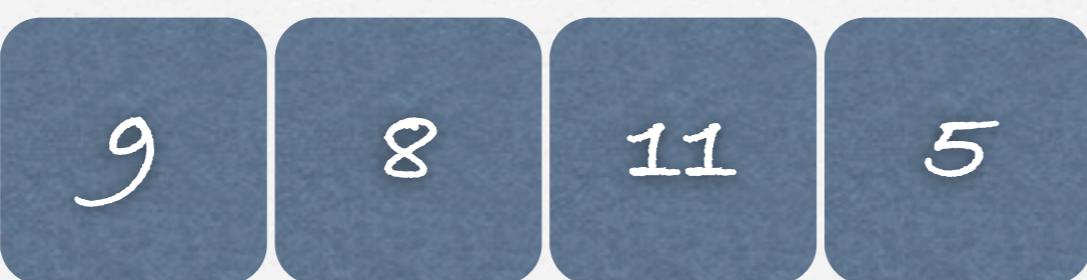
```
+++++[>+++++>++++++>++++  
+<<<-]>. >++. ++++++. .+++. >-.  
-----.<+++++++.-----.++  
+. -----.>+.
```



ptr

5

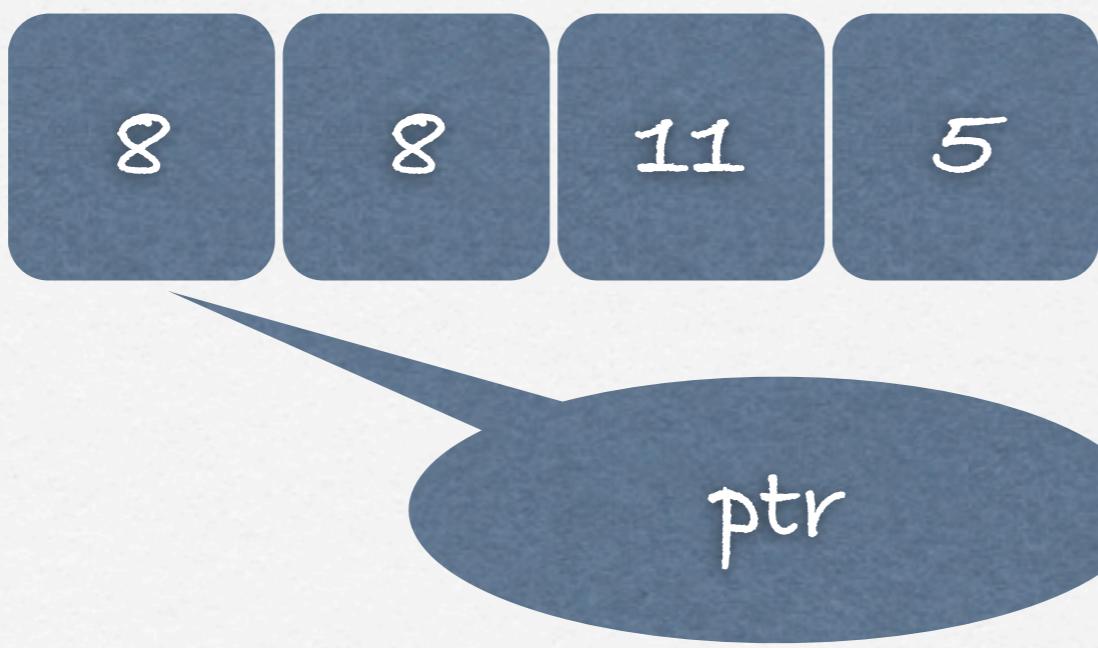
```
+++++[>+++++>++++++>++++  
+<<<-]>. >++. ++++++. .+++. >-.  
-----.<+++++++.-----.++  
+. -----.<-----.>+.
```



ptr

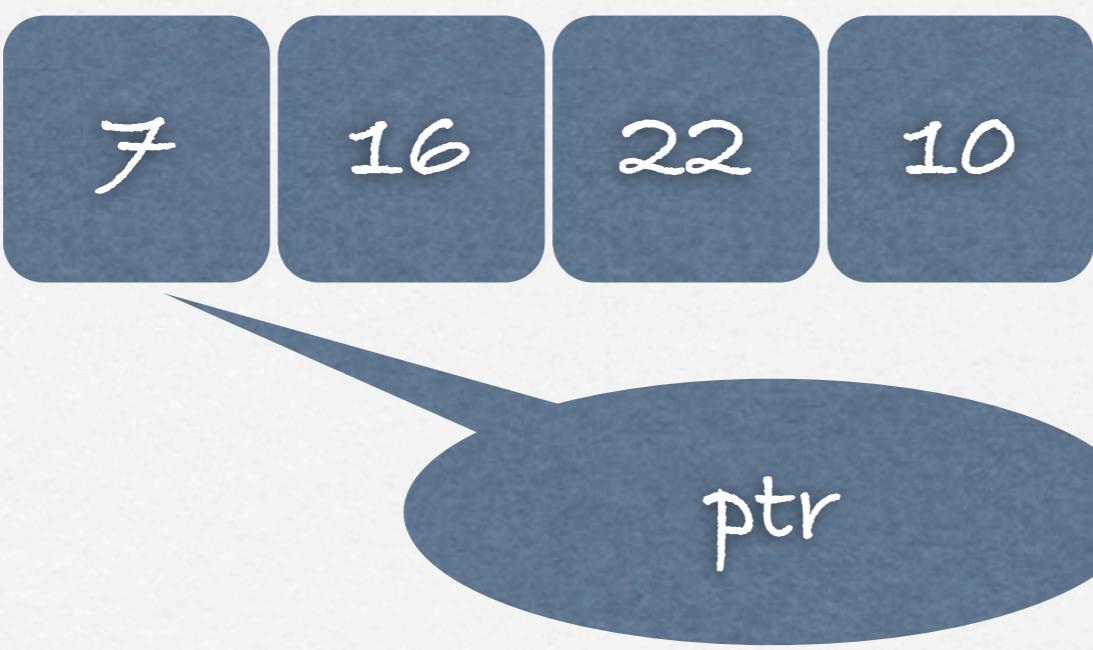
9 → 8

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



繰り返し

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



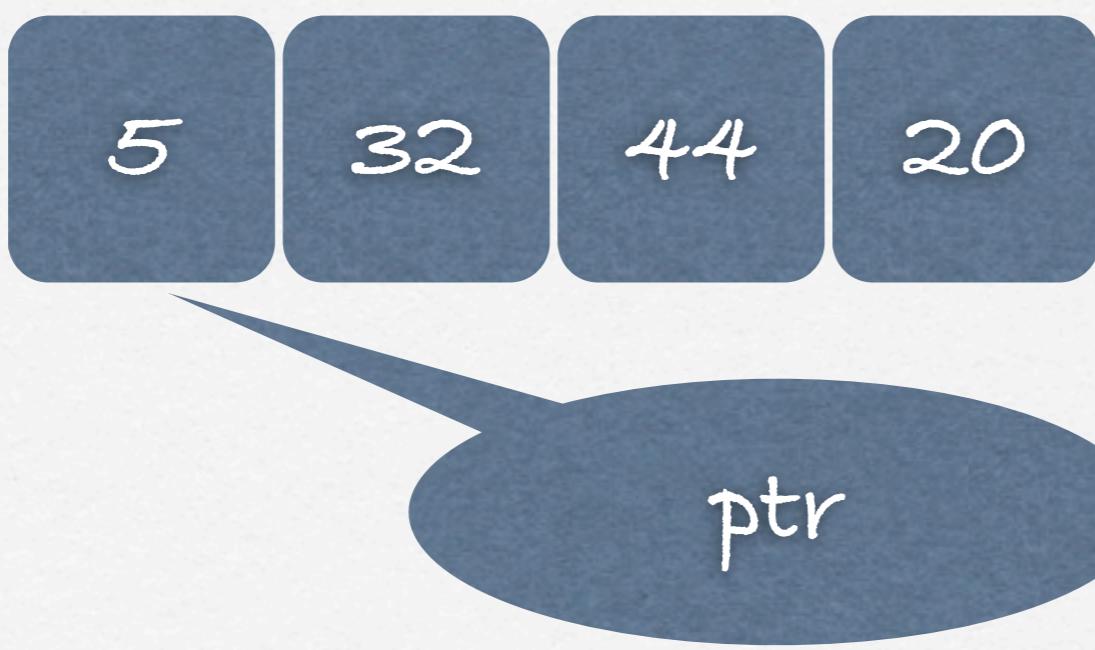
繰り返し

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



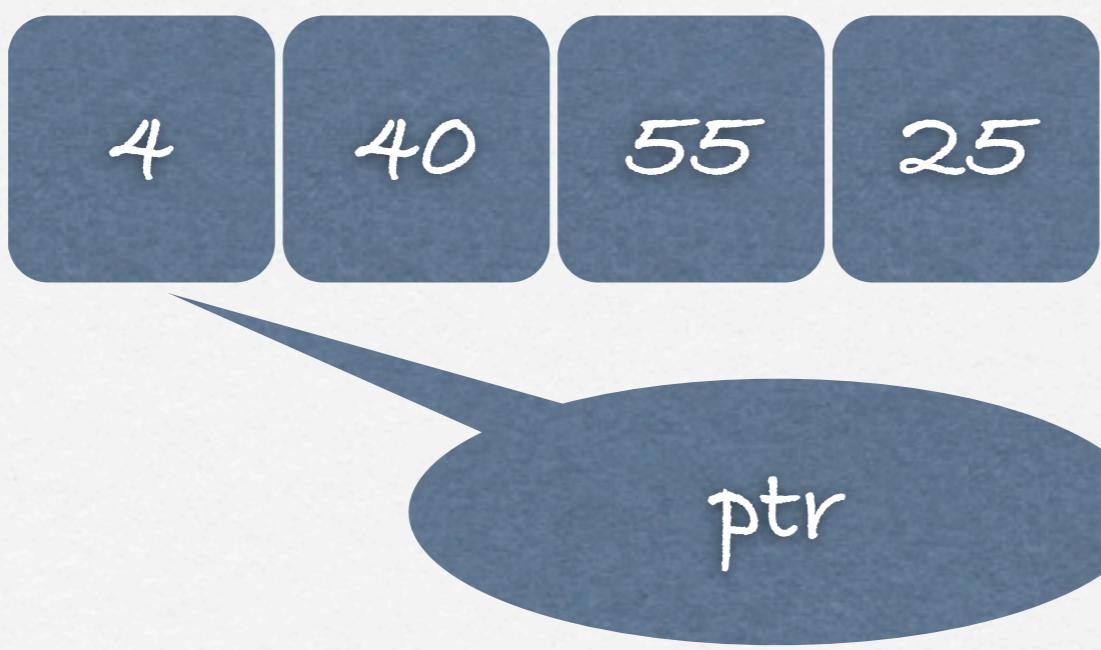
繰り返し

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



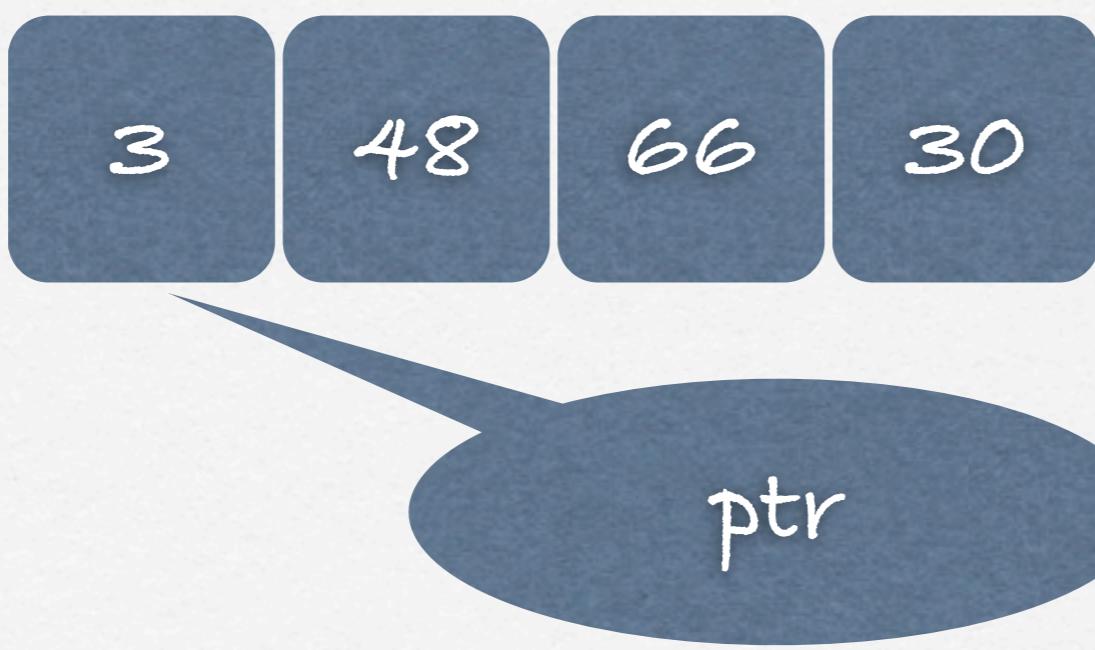
繰り返し

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



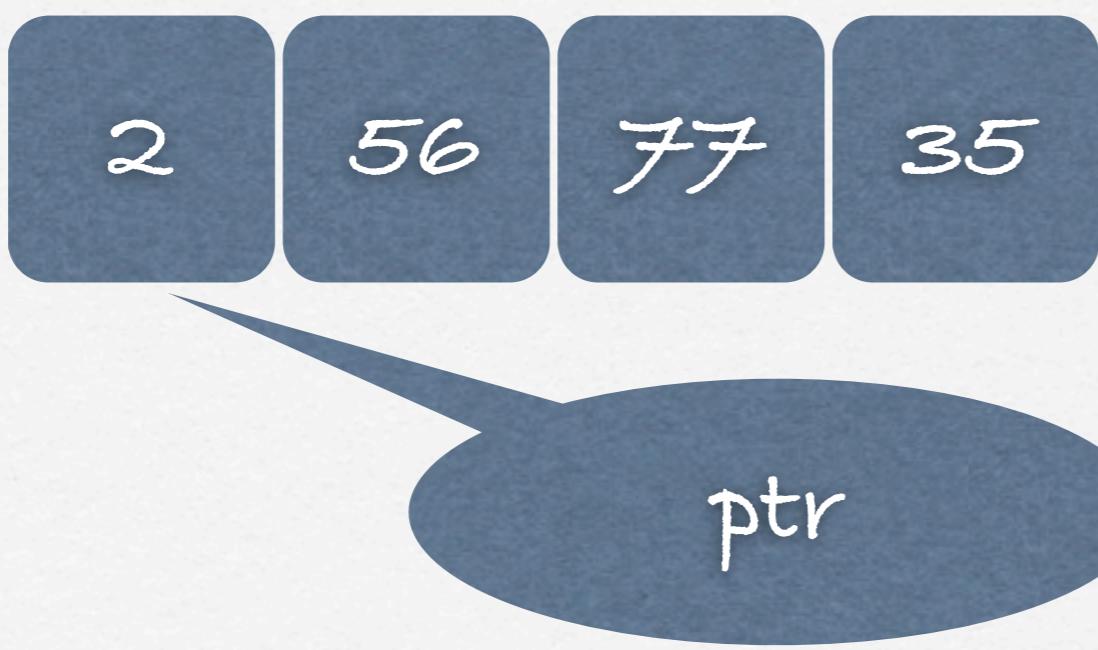
繰り返し

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



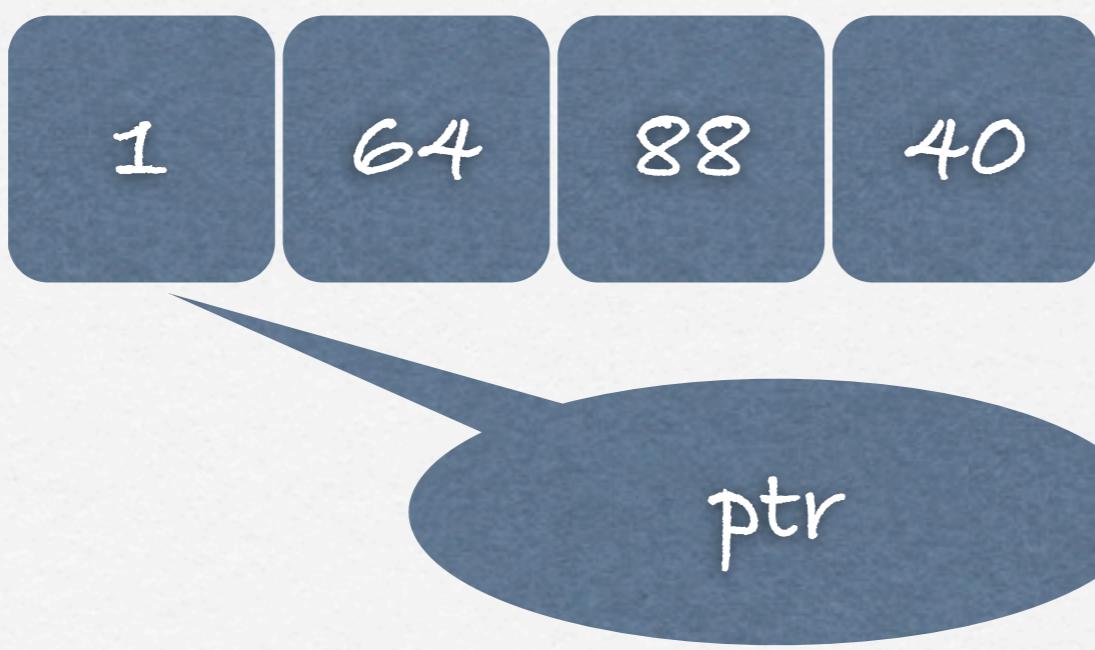
繰り返し

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



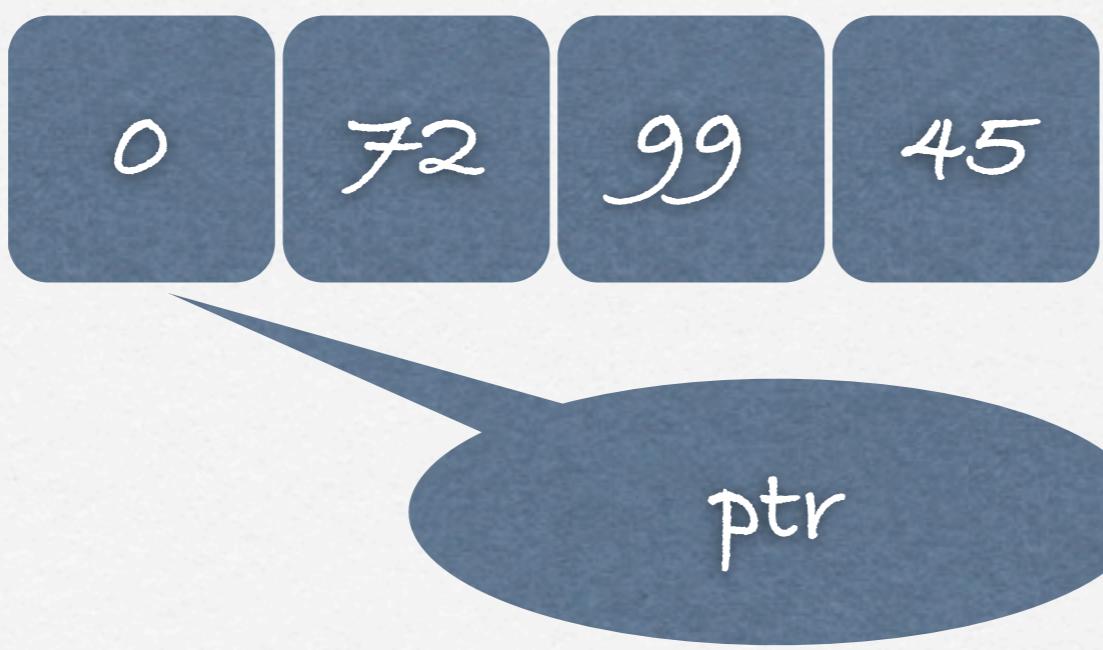
繰り返し

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



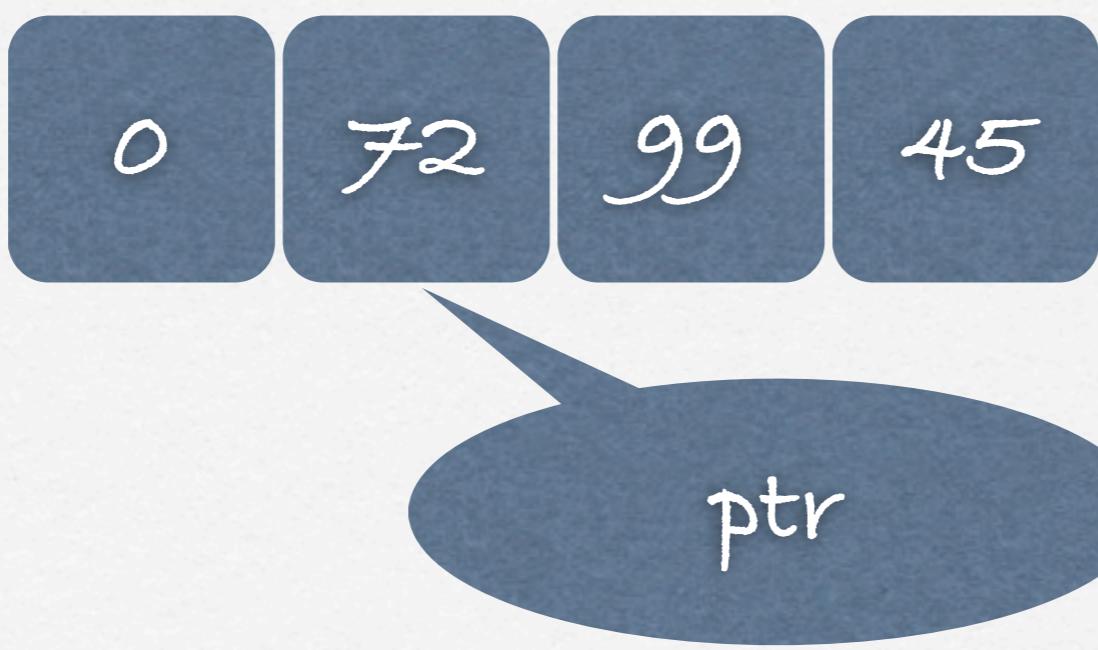
繰り返し

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



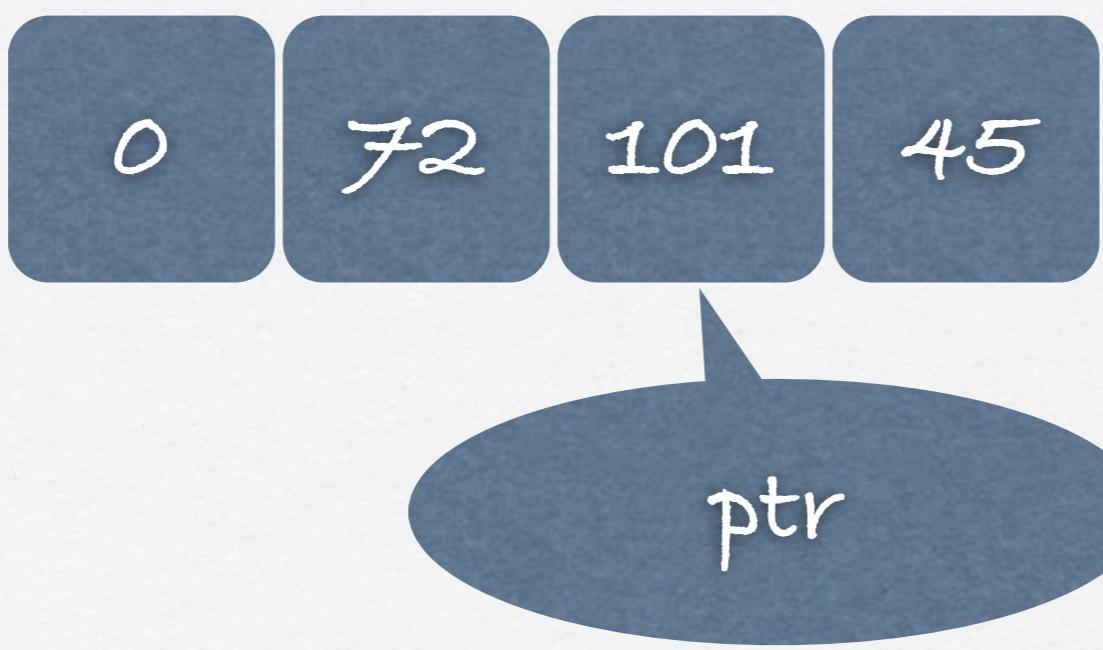
“H”

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



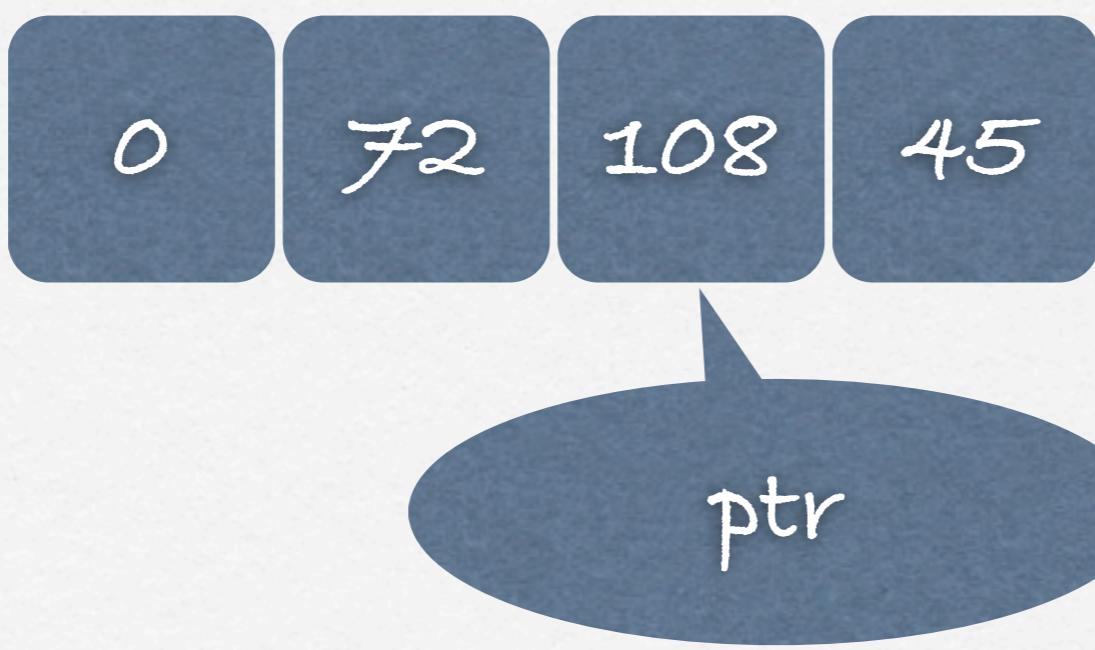
“e”

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



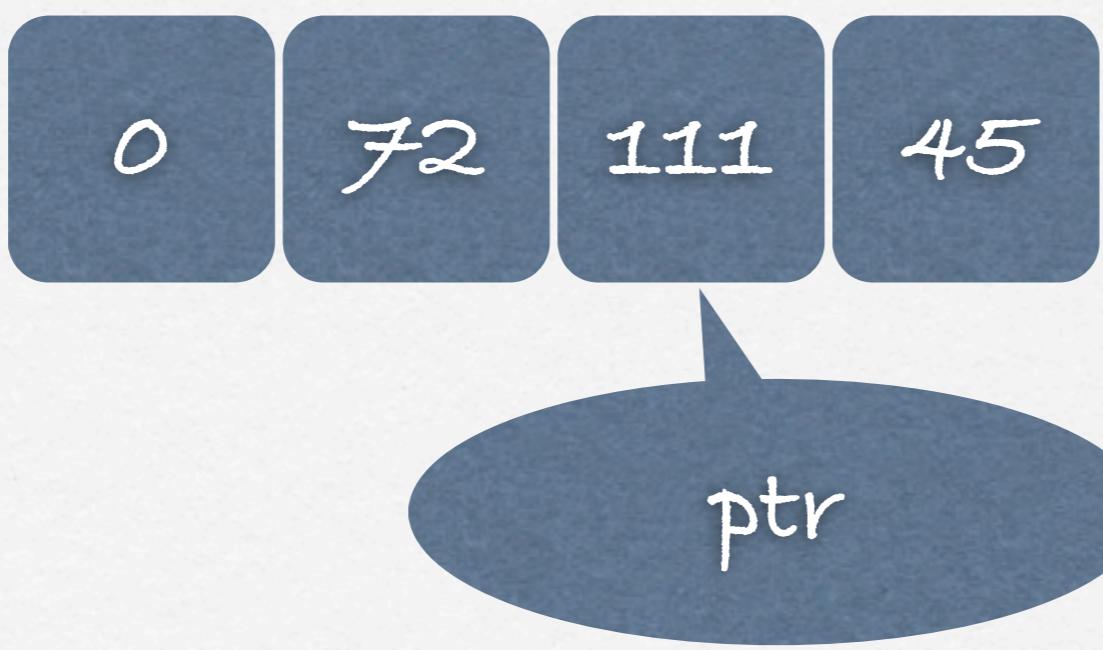
“|”, “|”

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



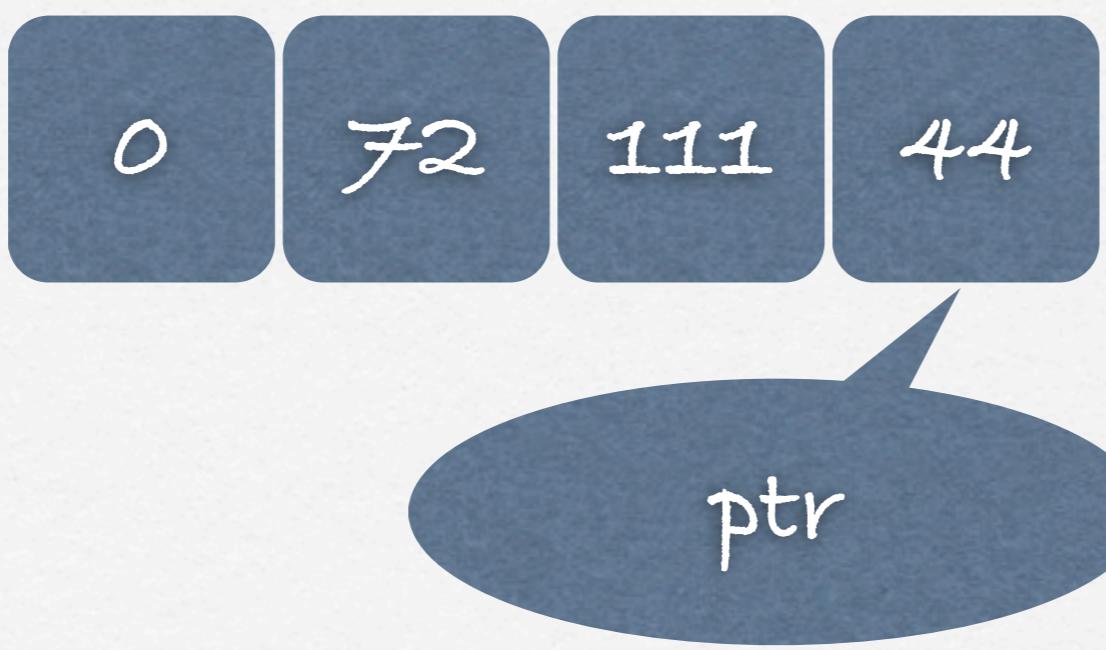
“O”

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



“ ”
‘

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



やっと“Hello,”

The image shows a single, blank page from a spiral-bound notebook. The page is white with horizontal blue ruling lines. It is held in place by a black metal spiral binding along the top edge. The right side of the page shows the dark blue cover of the notebook.

続きはWebで

本題

- Brainfuck処理系を作ったよ
- シェルスクリプトで実装
- Brainfuck → 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

- “>”
 - $\text{PTR} = \$((\$PTR + 1))$
- “+”
 - $\text{VAL} = \text{VAL}_{\$PTR}$
 $\text{VAL} = \$((\$VAL))$
 $\text{eval } \text{VAL}_{\$PTR} = \$((\$VAL + 1))$

実装

- ":"
- VAL=VAL_\$PTR
VAL=\$((VAL))
OCH=`echo "obase=8; \$VAL" | bc`
OCH=`echo "\$OCH" | xxd -r -p`
/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

□ “I”

□ VAL=VAL_\$PTR

VAL=\$((VAL))

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

□ “I”

□ VAL=VAL_\$PTR

VAL=\$((VAL)); end

実行

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

デモ

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

まとめ

- 口 シェルはインタプリタ
- 口 動的に評価することができる
- 口 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！！！
- 口 ほかの言語処理系も実装できる？

Brainfush

- github.com
- <http://g.c./takano32/brainfush>
- [slideshare](https://www.slideshare.net/takano32/brainfush)
- <http://w.s.n./takano32/brainfush>

ご清聴ありがとうございました

