

春なので愛をこめて
HelloWorldを
つくって来た

2009年2月25日

すぎやーん(id:sugyan)

sugi1982@gmail.com

すぎやーん

(id:sugyan)

です

3回目の

Lightning Talks

伝説の焼肉
ヒッチハイク

プログラマ歴

もうすぐ半年

そるそる

技術ネタを

でかごとくといふ

“Hello World!”

を作ってみた

Hello World!

とは

標準出力に

“Hello World!”

と表示する

プログラム

基礎の基礎

精一杯の

愛をこめて

書いてみた

' '=~ (' (' . ' ?' . ' { ' . (' ` ' | ' % ') . (' [' ^ ' - ') . (' ` ' | ' ! ') . ((' ` ') | ' , ') . ' " ' . (' [' ^ ' + ') . (' [' ^ ') ') . (' ` ' | ') ') . (' ` ' | ' . ') . (' [' ^ ' / ') . (' { ' ^ ' [') . (' \ \ ') . ' " ' . (' ` ' ^ ' (') . (' ` ' | ' % ') . (' ` ' | ' , ') . (' ` ' | ' , ') . (' ` ' | ' / ') . (" \ { " ^ ' [') . (' [' ^ ' , ') . (' ` ' | ' / ') . ((' [') ^ ') ') . (' ` ' | ' , ') . (' ` ' | ' \$ ') . ' ! ' . ' \ \ ' . ' \ \ ' . (' ` ' | ' . ') . ' \ \ ' . ' " ' . ' " ' . " \ } " . ') ') ; \$:= ' . ' ^ ' ~ ' ; \$ ~ = ' @ ' | ' (' ; \$ ^ = ') ' ^ ' [' ; \$ / = ' ` ' | ' . ' ; \$, = ' (' ^ ' } ' ; \$ \ = (' ` ') | ' ! ' ; \$:= ') ' ^ ' } ' ; \$ ~ = ' * ' | ' ` ' ; \$ ^ = ' + ' ^ ' _ ' ; \$ / = ' & ' | ' @ ' ; \$, = ' [' & ' ~ ' ; \$ \ = ' , ' ^ " \ | " ; \$:= ' . ' ^ ' ~ ' ; \$ ~ = ' @ ' | ' (' ; \$ ^ = ') ' ^ ' [' ; \$ / = ' ` ' | ' . ' ; \$, = ' (' ^ ' } ' ; \$ \ = ' ` ' | (' ! ') ; \$:= ') ' ^ ' } ' ; \$ ~ = ' * ' | ' ` ' ; \$ ^ = ' + ' ^ ' _ ' ; \$ / = ' & ' | ' @ ' ; \$, = ' [' & ' ~ ' ; \$ \ = ' , ' ^ ' | ' ; \$:= " \ . " ^ ' ~ ' ; \$ ~ = ' @ ' | ' (' ; \$ ^ = ') ' ^ ' [' ; \$ / = ' ` ' | ' . ' ; \$, = ' (' ^ ' } ' ; \$ \ = ' ` ' | ' ! ' ; \$:= ') ' ^ ' } ' ; \$ ~ = ' * ' | ' ` ' ; \$ ^ = ' + ' ^ ' _ ' ; \$ / = ' & ' | ' @ ' ; \$, = ' [' & ' ~ ' ; \$ \ = ' , ' ^ ' | ' ; \$:= ' . ' ^ ' ~ ' ; \$ ~ = ' @ ' | ' (' ; \$ ^ = ') ' ^ ' [' ; \$ / = ' ` ' | ' . ' ; \$, = ' (' ^ ' } ' ; \$ \ = ' ` ' | ' ! ' ; \$:= " \) " ^ ' } ' ; \$ ~ = ' * ' | ' ` ' ; \$ ^ = ' + ' ^ ' _ ' ; \$ / = ' & ' | ' @ ' ; \$, = ' [' & ' ~ ' ; \$ \ = (' , ') ^ ' | ' ; \$:= ' . ' ^ ' ~ ' ; \$ ~ = ' @ ' | ' (' ; \$ ^ = ') ' ^ ' [' ; \$ / = ' ` ' | (' . ') ; \$, = ' (' ^ ' } ' ; \$ \ = ' ` ' | ' ! ' ; \$:= ') ' ^ ' } ' ; \$ ~ = ' * ' | ' ` ' ; \$ ^ = ' + ' ^ ' _ ' ; \$ / = ' & ' | ' @ ' ; \$, = ' [' & ' ~ ' ; \$ \ = ' , ' ^ ' | ' ; \$:= ' . ' ^ ' ~ ' ; \$ ~ = ' @ ' | ' (' ; \$ ^ = ') ' ^ ' (' [') ; \$ / = ' ` ' | ' . ' ; \$, = ' (' ^ ' } ' ; \$ \ = (' ` ') | ' ! ' ; \$:= ') ' ^ ' } ' ; \$ ~ = ' * ' | ' ` ' ; \$ ^ = " \ + " ^ _ ' ; \$ / = ' & ' ;

フォントが
が小さい？

もうちよつと

簡潔に書く

' '=~ (' (? { ' . (
' [)) @ [] ^ ` [@ @ @ ~ , @ [@ ? ~ ! . _ ' ^
' + [@ . / } | (> , , / ^ { /) , [_ } @ } '
) . ' }) ')

Perlの

プログラムです

記号で作る

プログラム

可読性: 0

実用性: 0

なぜ記号だけで
プログラムが
動くのか？

キーワード

eval

擴張正規表現

文字列論理演算

eval

文字列を
プログラム
として解釈

```
$a='print"Hello World!\n";  
eval $a;
```

擴張正規表現

(?{ code })

正規表現の構文
で eval する

```
' ' = ~ (' (?{ ' .
```

```
' print "Hello World! \n" ' .
```

```
' } ) ' )
```

文字列の 論理演算

ASCII

文字コード

'p' == > 0x70

== > 01110000

排他的論理和

XOR

$$0 \wedge 0 = 0$$

$$0 \wedge 1 = 0$$

$$1 \wedge 0 = 0$$

$$1 \wedge 1 = 1$$

$$\text{!} [\text{!} \equiv \text{!} > 01011011$$

$$\text{!} + \text{!} \equiv \text{!} > 00101011$$

' [' ⇒ > 0 1 0 1 1 0 1 1

' + ' ⇒ > 0 0 1 0 1 0 1 1

0 1 1 1 0 0 0 0

'p' => 01110000

(' [' ^ ' + ') eq ' p '

(') ' ^ ' [') eq ' r '

(') ^ ' @ ') eq ' i '

('@ ^ . ') eq ' n '

(' [' ^ ' / ') eq ' t '

(' [)) @ [' ^

' + [@ . / ') eq

' print '

' '=~ (' (? { ' . (
 ' [)) @ [] ^ ` [@ @ @ ~ , @ [@ ? ~ ! . _ ' ^
 ' + [@ . / } | (> , , / ^ { /) , [_ } @ } '
) . ' }) ')

できあがり

記号プログラム
生成モジュール

Acme::EyeDrops

プログラムを
任意の形にして
記号で出力


```
#!/usr/bin/perl
use strict;
use warnings;
use Acme::EyeDrops qw(sightly);

my $shape = `banner -w 36 Love`;
my $source = 'print "Hello world!\n"';
print sightly({
    ShapeString => $shape,
    SourceString => $source,
    Regex      => 1,
    Rotate     => 270,
});
```


原理さえ

知ってしまえば

自分で作ることも

出来るはず

ワンライナー記号化 スクリプトを書いた

<http://d.hatena.ne.jp/sugyan/20090206/1233852075>

```
$ perl -e '"!~(" (?{" . ("+,@.) ]} (>, , / ^ _ @ /  
@?|!@^"^[^)@]}_` [@@@~ (/] , [ ] } . | " ) . " } ) " ) '  
Hello world!
```

その他のネタ

予約語

プログラミング


```
#!/usr/bin/perl
length q chr uc and print chr ord uc
q chop uc and print chr ord q ne sin
and print chr ord qw q le q and print
chr ord q else and print chr ord q
pop and print chr ord qq q q and
print chr ord uc q qw eq and print
chr ord q local and print chr ord q
qr q and print chr ord q else and
print chr ord qw q do q and print chr
hex length q q bless localtime ref q
and print chr length q continue vec
```

ppencode

<http://www.namazu.org/~takesako/diary/?date=20050831>

JAPH

http://en.wikipedia.org/wiki/Just_another_Perl_hacker

Code Golf

<http://codegolf.com/>

他にも何か
ネタがあれば
教えて下さい

Rubyも

すごいプログラムを

書けるらしい

これから
勉強します

><

enjoy programming!!