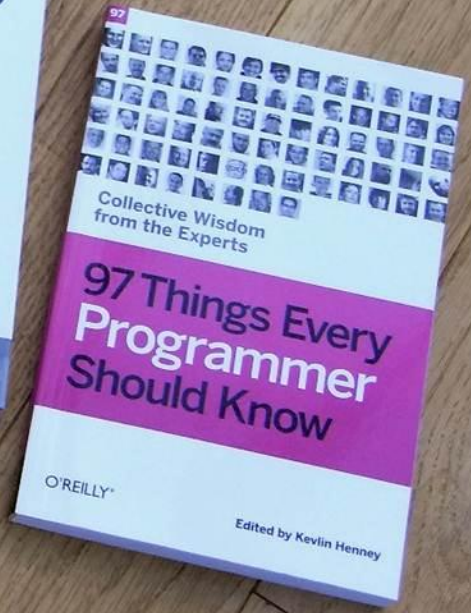
Cool & Useless

@KevlinHenney


WILEY SERIES IN
SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
A Pattern Language for Distributed Computing
Volume 4
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt


WILEY SERIES IN
SOFTWARE DESIGN PATTERNS
PATTERN-ORIENTED SOFTWARE ARCHITECTURE
On Patterns and Pattern Languages
Volume 5
Frank Buschmann
Kevlin Henney
Douglas C. Schmidt


Collective Wisdom from the Experts
97 Things Every Programmer Should Know
O'REILLY
Edited by Kevlin Henney

cool, *adjective*

- fashionably attractive or impressive
- excellent
- used to express acceptance or agreement
- unfriendly or unenthusiastic
- used as an intensive
- used when a conversation goes silent
- marked by deliberate effrontery or lack of due respect or discretion
- restrained or relaxed in style

Oxford English Dictionary · Concise Oxford English Dictionary · Merriam-Webster's Collegiate Dictionary · Urban Dictionary

```
STR
16571  LD C A      79
(187)  LD L A      185
( 64)  LD H N      38 67
16575  LD A (HL)   126
(191)  LD B N      6  1
( 64)  AND N       230 127
       CP N        254 0
       JP Z DIS    40 20
       INC B       4
       CP N        254 110
       JR2DIS      40 15
       CP N        254 39
       JP C DIS    56 11
       LD A (HL)   126          '0' DIFF
       INC B       4
       LD L N      46 55        '1' EMPTY
       ADD (HL)    134
       BIT 7,A     203 127      '2' WALL
       JP Z DIS    40  2
       LD B N      6   0        '3' SAME
       LD A B      120
       LD L C      105
       RET         201
TABLES
16607  1 11 -1 -11 -10 -12 12 10
16615  13 -13 21 -21 23 -23 -9 9
16623  11 10 12
16626  54 55 39 51 53
PIECE
16631  XOR A       175
(247)  LD(NN) A    50 70 64
( 64)  LD A (HL)   126
       AND N       230 127
       CP P        254 53
       JP Z DIS    40 79
       LD C N      14 1
       LD B N      6  8
       LD HL NN    33 231 64
       CP N        254 51
       JP Z DIS    40 22
       LD L N      46 223
       CP K        254 48
       JP Z DIS    40 16
       LD C B      72
       CP H        254 54
       JP Z DIS    40 11
       LD B N      6  4
       CP R        254 55
       JP Z DIS    40  5
       LD L N      46 227
       CP B        254 39
       RET NZ      192

SHIFT
16882  LD HL NN    33 99 64
(242)  LD DE NN    17 70 64
( 65)  LD BC NN    1  28 0
       JP C DIS    56 1
       EX DE HL    235
       LDIR        237 176
       RET         201
PSC
17162  AND N       230 127
(10)   LD HL NN    33 242 64
(67)   LD B  N      6  5
       CP (HL)     190
       RET Z       200
       INC HL      35
       DJNZ DIS    16 251
       LD A B      120
       RET         201
MPSCAN
17046  XOR A       175
(150)  LD (NN) A   50 65 64
( 66)  LD B N      6  86
       LD HL NN    33 62 67
       INC HL      35
       PUSH HL     229
       PUSH BC     197
       LD E L      93
       CALL STR2   205 191 64
       CP N        254 3
       JP NZ DIS   32 41
       LD L E      107
       LD (NN) HL  34 7 64
       CALL MOVE   205 247 64
       CALL TL     205 130 66
       JP Z DIS    40 29
       LD E A      95
       LD D N      22 67
       CALL PMOVE  205 255 66
       EXX         217
       AND A       167
       CALL SHIFT  205 242 65
       CALL CHK    205 1 66
       EXX         217
       LD(HL) B    112
       LD A C      121
       LD(DE) A    18
       JP C DIS    56 3
       CALL SCORE  205 153 65
       SCF         55
       CALL SHIFT  205 242 65
       JP DIS      24 222
       POP BC      193
       POP HL      225
       DJNZ DIS    16 200
       LD A (NN)   58 65 64
       CP N        254 0
       JP Z DIS    40 254 •
       LD HL NN    33 69 64
       LD A (HL)   126
       DEC HL      43
       DEC HL      43
       LD E (HL)   94
       LD D N      22 67
       LD (DE) A   18
       DEC HL      43
       LD L (HL)   110
       LD H D      98
17131  BIT 0,L     203 69
(235)  LD(HL) N    54 0
( 66)  JP Z DIS    40 2
       LD(HL) N    54 128
       CALL CHGMV  205 247 66
       RET         201
INC
17176  LD A L      125
(24)   EXX         217
(67)   LD(NN) A    50 128 64
       CALL SQ.AT  205 16 66
       EXX         217
       LD A C      121
       RET         201
DRIVER
16959  LD B N      6  5
(63)   LD A N      62 8
(66)   LD HL NN    33 159 67
       INC HL      35
       LD(HL) A    119
       DJNZ DIS    16 252
       CALL KT     205 160 64
       CP N        254 3
       JP NZ DIS   32 238
       LD (NN) HL  34 7 64
       LD E L      93
       CALL MOVE   205 247 64
       LD HL NN    33 161 67
       CALL KT     205 160 64
       CP N        245 2
       EX DE HL    235
       JP NC DIS   48 220
       CALL TL     205 130 66
       JP Z DIS    40 215
       CP C        185
       JP NZ DIS   32 248
       CALL PMOVE  205 255 66
       EXX         217
       CALL CHK    205 1 66
       EXX         217
       JP C DIS    56 8
       CALL CHGSQ  205 235 66
       CALL MPSCAN 205 150 66
       JP DIS      24 194
       LD (HL) B   112
       LD A C      121
       LD (DE) A   18
       JP DIS      24 249
TEST LIST
17026  LD HL NN    33 70 64
(130)  DEC (HL)    53
( 66)  LD A (HL)   126
       INC A       60
       RET Z       200
       ADD L       133
       LD L A      111
       LD A (HL)   126
       RET         201

PAWN
16721  LD A (HL)   126
(81)   AND N       230 128
(65)   LD HL NN    33 228 64
       JP NZ DIS   32 2
       LD L  N     46 241
       LD D  N     22 3
       LD A  E     123
       ADD (HL)    134
       PUSH HL     229
       PUSH AF     245
       CP N        254 63
       JP C DIS    56 32
       CP N        254 148
       JP NC DIS   48 28
       CALL STR    205 187 64
       CP N        254 0
       JP Z DIS    40 28
       CP N        254 1
       JP NZ DIS   32 17
       LD A D      122
       CP N        254 1
       JP NZ DIS   32 12
       CALL ALIST  205 141 66
       LD A E      123
       CP N        254 82
       JP C DIS    56 19
       CP N        254 126
       JP NC DIS   48 15
       POP AF      241
       POP HL      225
   DEC INC HL      43
       DEC D       21
       JP NZ DIS   32 210
       RET         201
       LD A D      122
       CP N        254 1
       CALL NZ ALIST 196 141 66
       JP DIS      24 241
       POP AF      241
       POP HL      225
       LD E A      95
       JP DIS      24 197

CHK
16897  LD A (NN)   58 55 67
( 1)   ADD N       198 48
(66)   LD HL NN    33 62 67
       LD B A      71
       CPIR        237 177
       DEC HL      43
       LD (NN) HL  34 128 64
SQ.AT  LD B N      6 86
16912  LD HL NN    33 62 67
(16)   INC HL      35
(66)   PUSH HL     229
       PUSH BC     197
       LD E L      93
       CALL STR2   205 191 64
       CP 0        254 0
       JP NZ DIS   32 25
       CALL CHEMV  205 247 66
       LD L E      107
       CALL MOVE   205 247 64
       CALL CHEMV  205 247 66
       CALL TL     205 130 66
       JP Z DIS    40 10
       LD HL(NN)   42 128 64

SCORE
16793  PUSH HL     229
(153)  PUSH BC     197
( 65)  PUSH DE     213
       PUSH HL     229
       PUSH BC     197
       LD D L      85
       LD HL NN    33 64 64
       CALL NN     205 36 7
       CALL PSC    205 10 67
       LD A B      120
       ADD A H     132
       LD C A      79
       POP AF      241
       CALL PSC    205 10 67
       POP HL      225
       CALL INC    205 24 67
       JP NC DIS   48 1
       ADD A B     128
       LD C A      79
       POP HL      225
       POP DE      209
       LD E (HL)   94
       LD (HL) D   114
       PUSH HL     229
       PUSH DE     213
       CALL INC    205 24 67
       JP NC DIS   48 1
       SUB B       144
       PUSH AF     245
       CALL CHEMV  205 247 66
       CALL CHK    205 1 66
       POP BC      193
       JP NC DIS   48 2
       INC B       4
       INC B       4
       POP DE      209
       POP HL      225
       LD (HL) E   115
       POP HL      225
       CALL CHG    205 250 66
       CALL INC    205 24 67
       JP NC DIS   48 1
       DEC B       5
       CALL CHG    205 250 66
       CALL CHEMV  205 247 66
       LD A B      120
       LD HL NN    33 60 64
       LD (HL) A   119
       EX DE HL    235
       LD HL NN    33 65 64
       CP (HL)     190
       RET C       216
       LD BC NN    1  5  0
       JP DIS      24 11
```

useless, *adjective*

- ineffectual
- serving no purpose
- destitute of useful qualities
- not answering or promoting the proposed desire or end
- having little ability or skill
- destitute of competence or capacity
- the end result of going through the American educational system

Oxford English Dictionary · Concise Oxford English Dictionary · Merriam-Webster's Collegiate Dictionary · Urban Dictionary

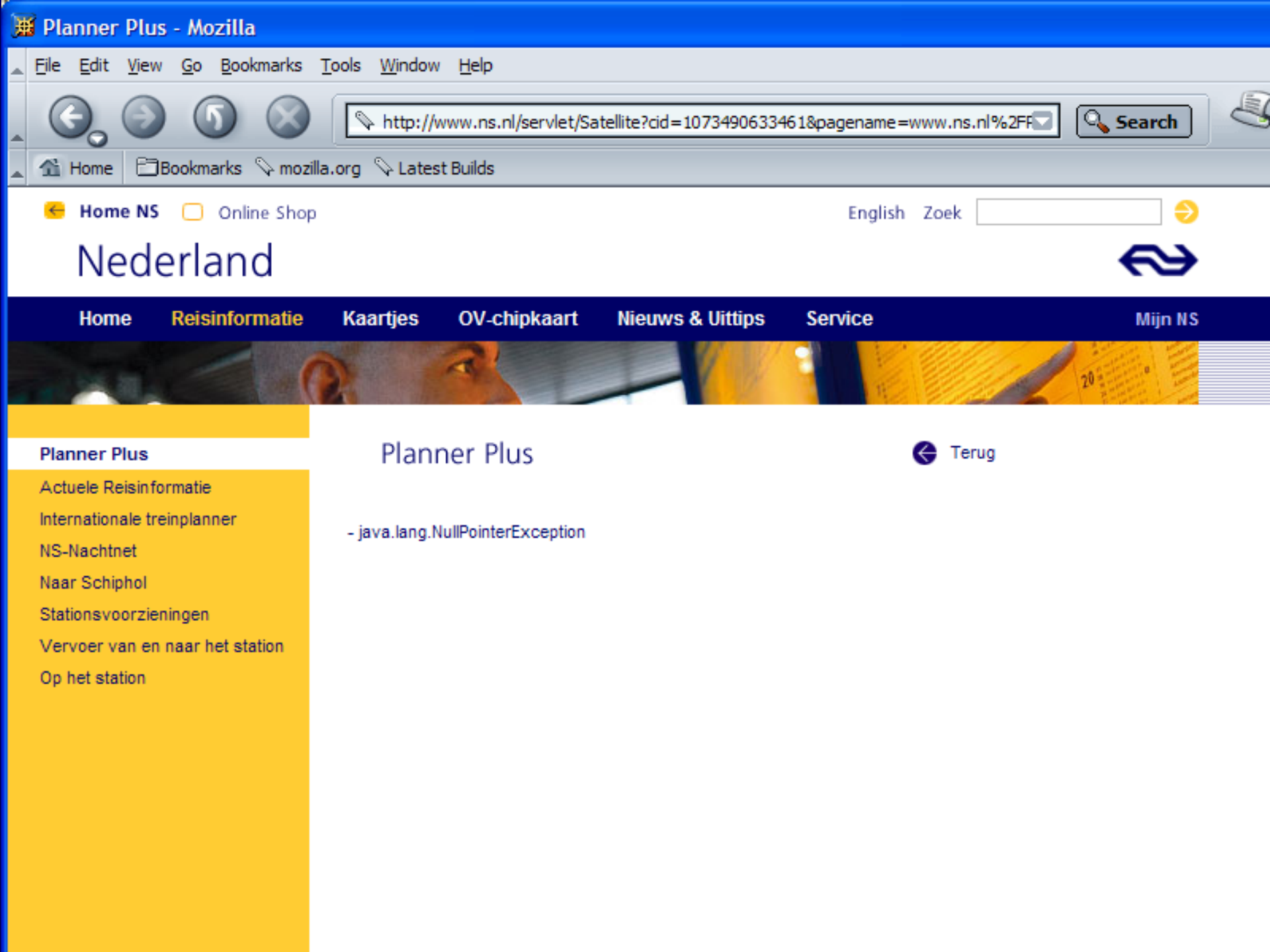
```
/*
 * Copyright (c) 1995, 2008, Oracle and/or its affiliates. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 *   - Redistributions of source code must retain the above copyright
 *     notice, this list of conditions and the following disclaimer.
 *
 *   - Redistributions in binary form must reproduce the above copyright
 *     notice, this list of conditions and the following disclaimer in the
 *     documentation and/or other materials provided with the distribution.
 *
 *   - Neither the name of Oracle or the names of its
 *     contributors may be used to endorse or promote products derived
 *     from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
 * IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

```
/*
 * Copyright (c) 1995, 2008, Oracle and/or its affiliates. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 *
 *   - Redistributions of source code must retain the above copyright
 *     notice, this list of conditions and the following disclaimer.
 *
 *   - Redistributions in binary form must reproduce the above copyright
 *     notice, this list of conditions and the following disclaimer in the
 *     documentation and/or other materials provided with the distribution.
 *
 *   - Neither the name of Oracle or the names of its
 *     contributors may be used to endorse or promote products derived
 *     from this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
 * IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT OWNER OR
 * CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
 * EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
 * PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
 * PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
 * NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

/**
 * The HelloWorldApp class implements an application that
 * simply prints "Hello World!" to standard output.
 */
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string.
    }
}
```

ampersand, *noun*

- the sign &, standing for *and*
- corruption of ‘and per se—and’, the old way of spelling *and* naming the character &, i.e., ‘& by itself is and’
- it was common practice to add the & sign at the end of the alphabet as the 27th letter, thus the recitation of the alphabet would end in ‘X, Y, Z and per se and‘

Oxford English Dictionary · Concise Oxford English Dictionary · Wikipedia


Planner Plus - Mozilla
File Edit View Go Bookmarks Tools Window Help
http://www.ns.nl/servlet/Satellite?cid=1073490633461&pagename=www.ns.nl%2FF
Search
Home Bookmarks mozilla.org Latest Builds
Home NS
Online Shop
English
Zoek
Nederland
Home
Reisinformatie
Kaartjes
OV-chipkaart
Nieuws & Uittips
Service
Mijn NS
Planner Plus
Actuele Reisinformatie
Internationale treinplanner
NS-Nachtnet
Naar Schiphol
Stationsvoorzieningen
Vervoer van en naar het station
Op het station
Planner Plus
Terug
- java.lang.NullPointerException


Cité
de
l'espace
arianespace
esa
esa
cnes
cnes

RUD, *noun*

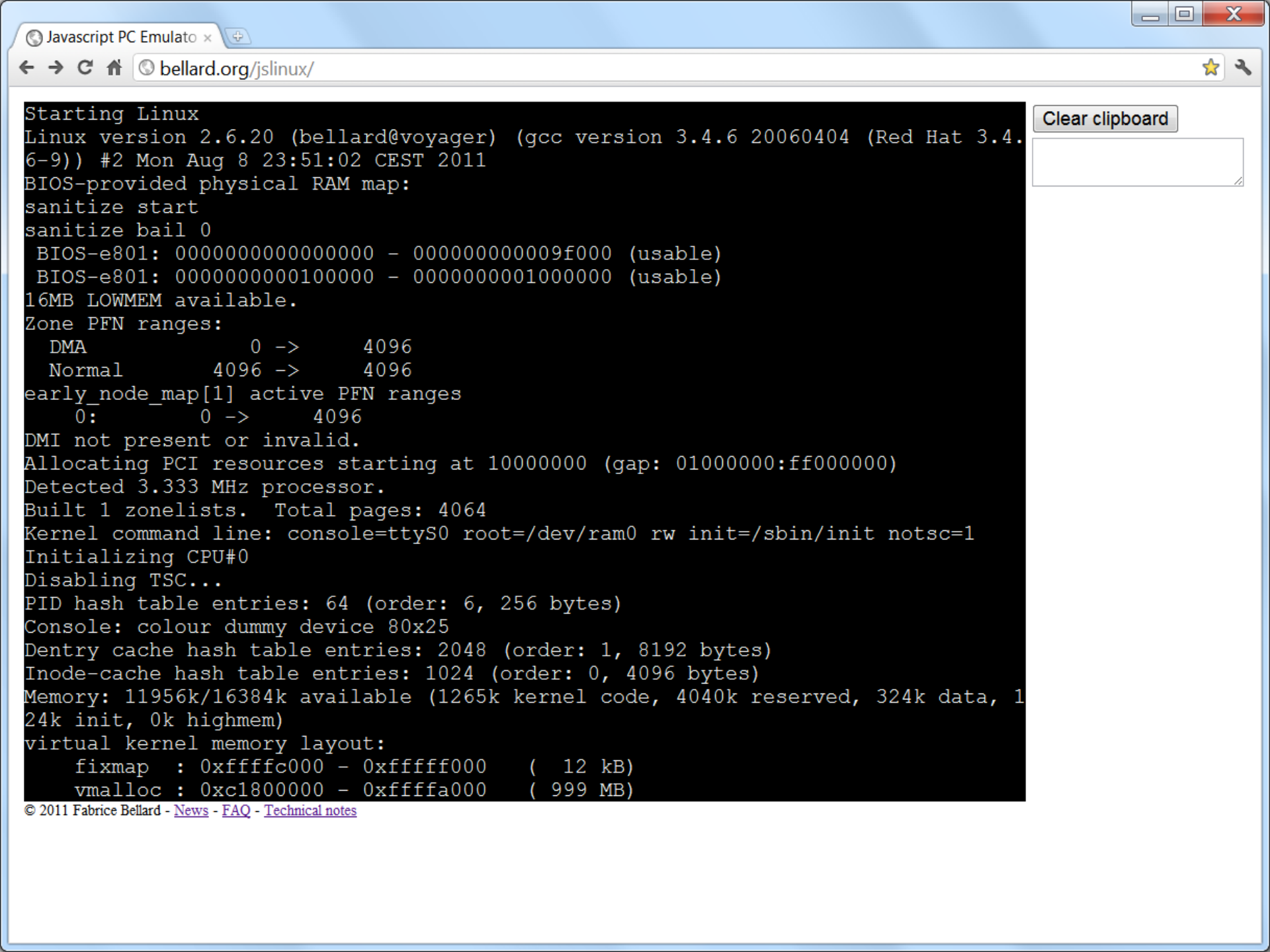
- Rapid Unscheduled Disassembly
- Rocket science and amateur rocketry jargon that's acronymous, euphemistic and explosively self-explanatory

```
      end if;
      L_M_DON_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_DON) *
                                      G_M_INFO_DERIVE(T_ALG.E_DON));
      if L_M_DON_32 > 32767 then
         P_M_DERIVE(T_ALG.E_DON) := 16#7FFF#;
      elsif L_M_DON_32 < -32768 then
         P_M_DERIVE(T_ALG.E_DON) := 16#8000#;
      else
         P_M_DERIVE(T_ALG.E_DON) := UC_16S_EN_16NS(
            TDB.T_ENTIER_16S(L_M_DON_32));
      end if;

      P_M_DERIVE(T_ALG.E_DOE) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
                                      ((1.0/C_M_LSB_DOE) *
                                      G_M_INFO_DERIVE(T_ALG.E_DOE)

      L_M_BV_32 := TDB.T_ENTIER_32S ((1.0/C_M_LSB_BV) *
                                      G_M_INFO_DERIVE(T_ALG.E_BV));
      if L_M_BV_32 > 32767 then
         P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
      elsif L_M_BV_32 < -32768 then
         P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
      else
         P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M
      end if;

      P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS (TDB.T_ENTIER_16S
                                      ((1.0/C_M_LSB_BH) *
                                      G_M_INFO_DERIVE(T_ALG.E_BH)))
   end LIRE_DERIVE;
--$finprocedure

   --(
   procedure LIRE_SEUIL (P_M_SEUIL : out TDB.T_ENTIER_16NS) is
      --\
```


(c) <DATE OVERFLOW>
POWERED BY COBOL ON COGS

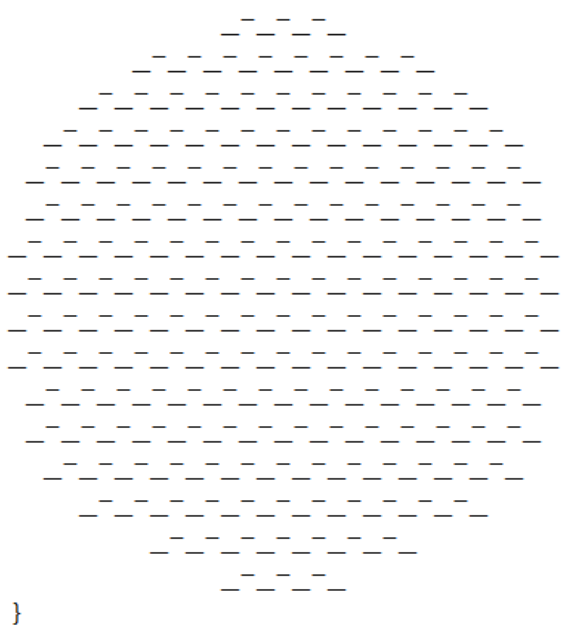
Any application that *can* be written in JavaScript, *will* eventually be written in JavaScript.

Atwood's Law

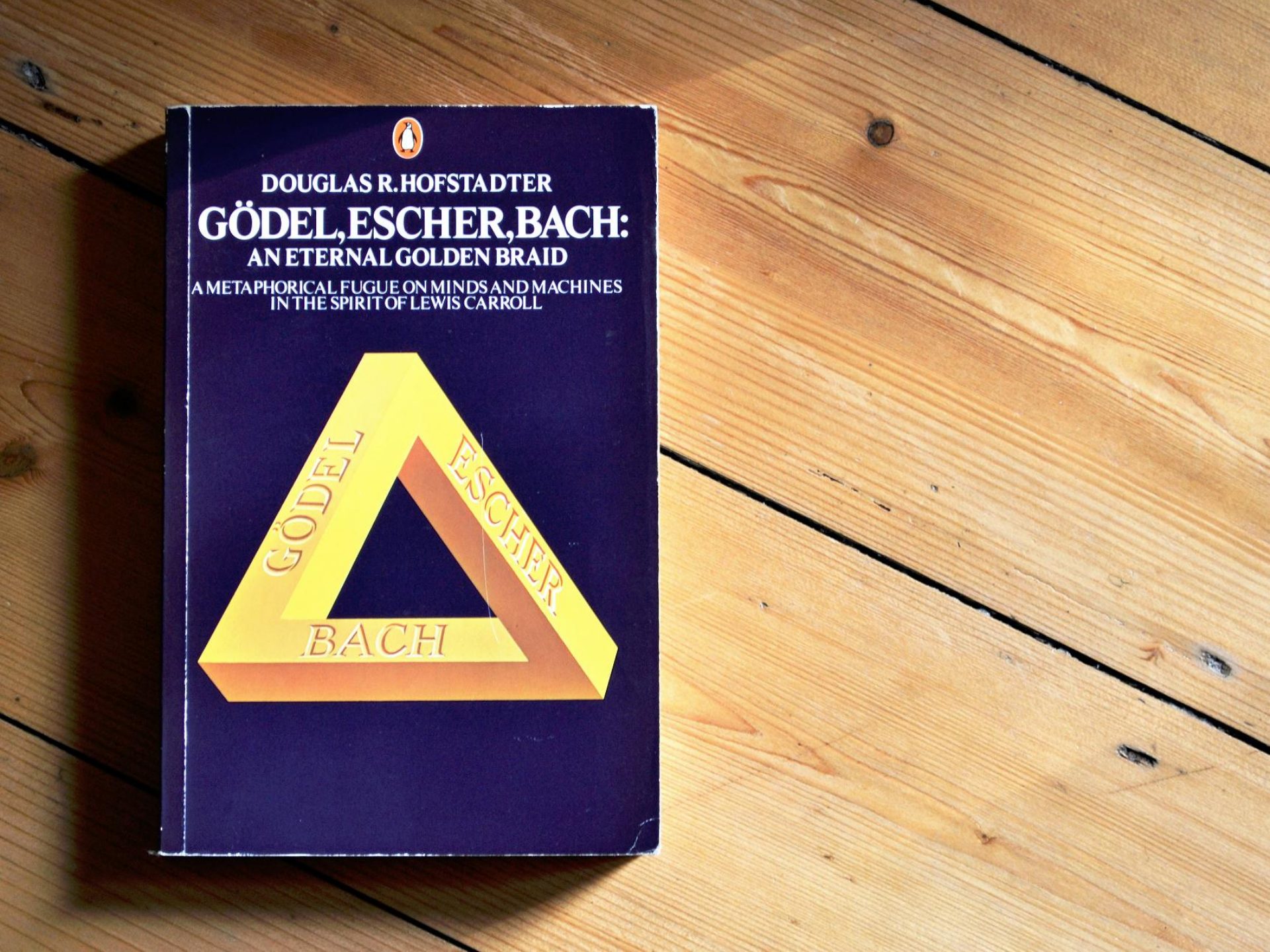

Javascript PC Emulato
bellard.org/jslinux/
Starting Linux
Linux version 2.6.20 (bellard@voyager) (gcc version 3.4.6 20060404 (Red Hat 3.4.
6-9)) #2 Mon Aug 8 23:51:02 CEST 2011
BIOS-provided physical RAM map:
sanitize start
sanitize bail 0
BIOS-e801: 0000000000000000 - 000000000009f000 (usable)
BIOS-e801: 0000000000100000 - 0000000001000000 (usable)
16MB LOWMEM available.
Zone PFN ranges:
DMA 0 -> 4096
Normal 4096 -> 4096
early_node_map[1] active PFN ranges
0: 0 -> 4096
DMI not present or invalid.
Allocating PCI resources starting at 10000000 (gap: 01000000:ff000000)
Detected 3.333 MHz processor.
Built 1 zonelists. Total pages: 4064
Kernel command line: console=ttyS0 root=/dev/ram0 rw init=/sbin/init notsc=1
Initializing CPU#0
Disabling TSC...
PID hash table entries: 64 (order: 6, 256 bytes)
Console: colour dummy device 80x25
Dentry cache hash table entries: 2048 (order: 1, 8192 bytes)
Inode-cache hash table entries: 1024 (order: 0, 4096 bytes)
Memory: 11956k/16384k available (1265k kernel code, 4040k reserved, 324k data, 1
24k init, 0k highmem)
virtual kernel memory layout:
fixmap : 0xffffc000 - 0xfffff000 (12 kB)
vmalloc : 0xc1800000 - 0xffffa000 (999 MB)
Clear clipboard
© 2011 Fabrice Bellard - News - FAQ - Technical notes

```
                                                                    char
                                                                  _3141592654[3141
          ],__3141[3141];_314159[31415],_3141[31415];main(){register char*
      _3_141,*_3_1415, *_3__1415; register int _314,_31415,__31415,*_31,
     _3_14159,__3_1415;*_3141592654=__31415=2,_3141592654[0][_3141592654
    -1]=1[__3141]=5;__3_1415=1;do{_3_14159=_314=0,__31415++;for( _31415
   =0;_31415<(3,14-4)*__31415;_31415++)_31415[_3141]=_314159[_31415]= -
1;_3141[*_314159=_3_14159]=_314;_3_141=_3141592654+__3_1415;_3_1415=
__3_1415    +__3141;for                 (_31415 = 3141-
           __3_1415  ;                  _31415;_31415--
           ,_3_141 ++,                  _3_1415++){_314
           +=_314<<2 ;                  _314<<=1;_314+=
          *_3_1415;_31                   =_314159+_314;
          if(!(*_31+1)                   )* _31 =_314 /
          __31415,_314                   [_3141]=_314 %
          __31415 ;* (                   _3__1415=_3_141
         )+= *_3_1415                     = *_31;while(*
         _3__1415 >=                      31415/3141 ) *
         _3__1415+= -                     10,(*--_3__1415
        )++;_314=_314                     [_3141]; if ( !
        _3_14159 && *                     _3_1415)_3_14159
        =1,__3_1415 =                     3141-_31415;}if(
        _314+(__31415                      >>1)>=__31415 )
        while ( ++ *                       _3_141==3141/314
       )*_3_141--=0                        ;}while(_3_14159
       ) ; { char *                        __3_14= "3.1415";
       write((3,1),                        (--*__3_14,__3_14
       ),(_3_14159                          ++,++_3_14159))+
      3.1415926; }                          for ( _31415 = 1;
     _31415<3141-                           1;_31415++)write(
    31415% 314-(                            3,14),_3141592654[
  _31415    ] +                            "0123456789","314"
  [ 3]+1)-_314;                             puts((*_3141592654=0
,_3141592654))                               ;_314= *"3.141592";}
```

```
#define _ -F<00||--F-OO--;
int F=00,OO=00;main(){F_OO();printf("%1.3f\n",4.*-F/OO/OO);}F_OO()
{
```



```
}
```


DOUGLAS R. HOFSTADTER
GÖDEL, ESCHER, BACH:
AN ETERNAL GOLDEN BRAID
A METAPHORICAL FUGUE ON MINDS AND MACHINES
IN THE SPIRIT OF LEWIS CARROLL
GÖDEL
ESCHER
BACH

Achilles: What I was driving [illegible] that exact order.

Tortoise: Oh, now I understand what you meant when you asked me "What sort of word is that?" The answer is that a philosopher by the name of "Willard Van Orman Quine" invented the operation, so I name it in his honor. However, I cannot go any further than this in my explanation. Why these particular five letters make up his name—not to mention why they occur in this particular order—is a question to which I have no ready answer. However, I'd be perfectly willing to go and—

Achilles: Please don't bother! I didn't really want to know everything about Quine's name. Anyway, now I know how to quine a phrase. It's quite amusing. Here's a quined phrase:

"IS A SENTENCE FRAGMENT" IS A SENTENCE FRAGMENT.

It's silly but all the same I enjoy it. You take a sentence fragment, quine it, and lo and behold, you've made a sentence! A true sentence, in this case.

Tortoise: How about quining the phrase "is a king with no subject"?

Achilles: A king without a subject would be—

Tortoise: —an anomaly, of course. Don't wander from the point. Let's have quines first, and kings afterwards!

Achilles: I'm to quine that phrase, am I? All right—

"IS A KING WITH NO SUBJECT" IS A KING WITH NO SUBJECT.

It seems to me that it might make more sense if it said "sentence"

As a result of this research, I created a language called HQ9+. HQ9+ is a very simple language consisting of four operations: H, Q, 9, and +. These operations can be used to create any of the types of example programs described above. They work as follows:

H Prints "Hello, world!"
Q Prints the entire text of the source code file.
9 Prints the complete canonical lyrics to "99 Bottles of Beer on the Wall"
+ Increments the accumulator.

HQ9+ is very simple, but allows you to do some things that are very difficult in other languages. For example, here is a program that creates *four* -- count 'em, four -- copies of itself on the screen:

```
qqqq
```

This produces:

```
qqqq
qqqq
qqqq
qqqq
```

Wow! Wasn't that straightforward?


v=0000;eval$s=%q~d=%!^Lcf<LK8,
4ZojjV)O>qIH1/n[|2yE[>:ieC
yH?b[F^e7C/56j|pmRe+:)B
PPu01Avw^<IiQ=5$'D-y?
6ygIL8xI#LNz3v}T=4W
}RT5-iJbbG5P-DHB<.
$*are@b4U351Q-ug5
PFixrPvl&<p[]1IJ
y]0`_PstfUxOC(q
zcaAi?]^lCVYp!;
(;v=(v-($*+[45,
360)+"al$s=%q#{
126}";d.gsub!(/
require"zlib"||
d.map{|c|n=(n||
e=["%x"%n].pack
Inflate.inflate(
)[0];22.times{|y|
(y*2.0-21)/22)**(;
2-1).times{|x|u=(e+
90*x/w+v+90,90/w];s[(
32<<".:%#")[4*u.count((
s+";_ The Qlobe#{" "*18+ (
oh, 2010")}";exit~;_ The Qlobe
_@7gj*LJ=c5nM)Tp1g0%Xv.,S[<>YoP
"%.#% :::##" 97N-A&Kj_K_><wS5rtWk@*a+Y5
::########## O98(Zh)'Iof*nm.,$C5Nyt=
"## ##########" g6`YT+qLw9k^ch|K'),tc
#. .####:#######" lL27FZ0ij)7TQCI)P7u
:############" R,YvZ_rnv6ky-G+4U'
########################" 00x8RR%`Om7VDp4M5
##########:#### %#####" EGgDt8Lm#;bc4zS^
.##############:##% .## ." /,}.YOIFj(k&q_V
.################. #. " ;s="v=%04o;ev"%
:################% : "])[n=0].to_i;)%
######### " ;;"%c"%126+$s<<
#######% " |\s|".*"/,"");;
:######. " ;d=d.unpack"C*"
.######: . ")*90+(c-2)%91};
:###### #: " &&"H*";e=Zlib::
.#####% :: " &&e).unpack("b*"
%### " ;w=(Math.sqrt(1-(
.#% " ;2))*23).floor;(w*
")[y*z=360,z]*2;u=u[
" ;y*80)+120-w+x]=(""<<
" ;"0"))/u.size]}};puts\
"# :#######" ;"Copyright(C).Yusuke End\
Copyright(C).Yusuke Endoh, 2010

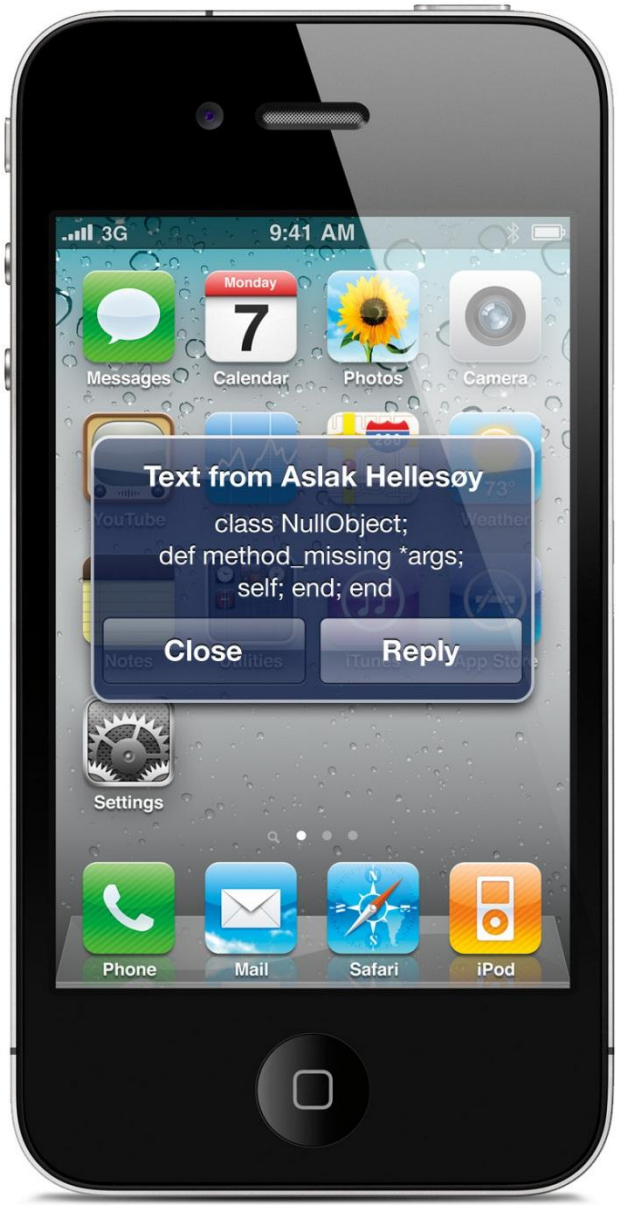
base / lib / base.rb

Edit this file

100644 | 91 lines (74 sloc) | 2.112 kb

raw | blame | history

```ruby
class Base
  VERSION = "0.0.2"

  def self.const_missing name
    all_modules.each do |mod|
      return mod.const_get(name) if mod.const_defined?(name)
    end
    super
  end

  def self.all_modules
    modules = ObjectSpace.each_object(Module).select do |mod|
      should_extract_from?(mod)
    end
    modules << Kernel
    modules
  end

  def self.should_extract_from?(mod)
    return false if module_is_a_base?(mod)
    return mod.is_a?(Module) && mod != Kernel
  end

  def self.method_missing name, *args, &block
    call_method(self, name, args, block) { super }
  end

  def method_missing name, *args, &block
    self.class.call_method(self, name, args, block) { super }
  end

  def self.call_method(object, name, args, block)
    name_string = name.to_s

    all_modules.each do |mod|
      if mod.respond_to?(name)
        return mod.send name, *args, &block
      elsif mod.instance_methods.include?(name_string)
        return call_instance_method(mod, name, args, block)
      end
    end

    # call "super" in the context of the method_missing caller
    yield
  end

  def self.call_instance_method(mod, name, args, block)
    if mod.is_a? Class
      klass = Class.new(mod)
    else
      klass = Class.new { include mod }
    end

    object = self.instantiate_regardless_of_argument_count(klass)
    return object.send name, *args, &block
  end

  def self.instantiate_regardless_of_argument_count(klass)
    (0..100).each do |arg_count|
      begin
        return klass.new(*[nil] * arg_count)
      rescue ArgumentError
      end
    end
  end

  def self.methods
    (giant_method_list_including_object(self) + super).uniq
  end

  def methods
    (self.class.giant_method_list_including_object(self) + super).uniq
  end

  # INHERIT ALL THE METHODS!
  def self.giant_method_list_including_object(object)
    methods = []
    all_modules.each do |mod|
      unless module_is_a_base?(mod)
        methods.concat(mod.methods).concat(mod.instance_methods)
      end
    end
    methods
  end

  def self.module_is_a_base?(mod)
    mod.is_a?(Base) || mod < Base || mod == Base
  end
end

```


3G
9:41 AM
Monday
7
Messages
Calendar
Photos
Camera
Text from Aslak Hellesøy
class NullObject;
def method_missing *args;
self; end; end
Close
Reply
Settings
Phone
Mail
Safari
iPod

```
for (Bar bar = foo as Bar; bar != null; bar = null)
{
    ...
}
```

```
/^1?$|^(11+?)\1+$/
```

```
// Erwin Unruh, untitled program,
// ANSI X3J16-94-0075/ISO WG21-462, 1994.

template <int i>
struct D
{
    D(void *);
    operator int();
};

template <int p, int i>
struct is_prime
{
    enum { prim = (p%i) && is_prime<(i>2?p:0), i>::prim };
};

template <int i>
struct Prime_print
{
    Prime_print<i-1>    a;
    enum { prim = is_prime<i,i-1>::prim };
    void f() { D<i> d = prim; }
};

struct is_prime<0,0> { enum { prim = 1 }; };
struct is_prime<0,1> { enum { prim = 1 }; };
struct Prime_print<2>
{
    enum { prim = 1 };
    void f() { D<2> d = prim; }
};

void foo()
{
    Prime_print<10> a;
}

// output:
// unruh.cpp 30: conversion from enum to D<2> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<3> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<5> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<7> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<11> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<13> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<17> requested in Prime_print
// unruh.cpp 30: conversion from enum to D<19> requested in Prime_print
```

```
:;while [ $? -eq 0 ];do nc -vlp 8080 -c'(r=read;e=echo;$r a b
c;z=$r;while [ ${#z} -gt 2 ];do $r z;done;f=`$e $b|sed 's/[^a-
zO-9_.-]//gi'`;h="HTTP/1.0";o="$h 200 OK\r\n";c="Content";if [
-z $f ];then($e $o;ls|(while $r n;do if [ -f "$n" ]; then $e
"<a href=\"/$n\">`ls -gh $n`</a><br>";fi;done););elif [ -f $f
];then $e "$o$c-Type: `file -ib $f`\n$c-Length: `stat -c%s
$f`";$e;cat $f;else $e -e "$h 404 Not Found\n\n404\n";fi)';done
```

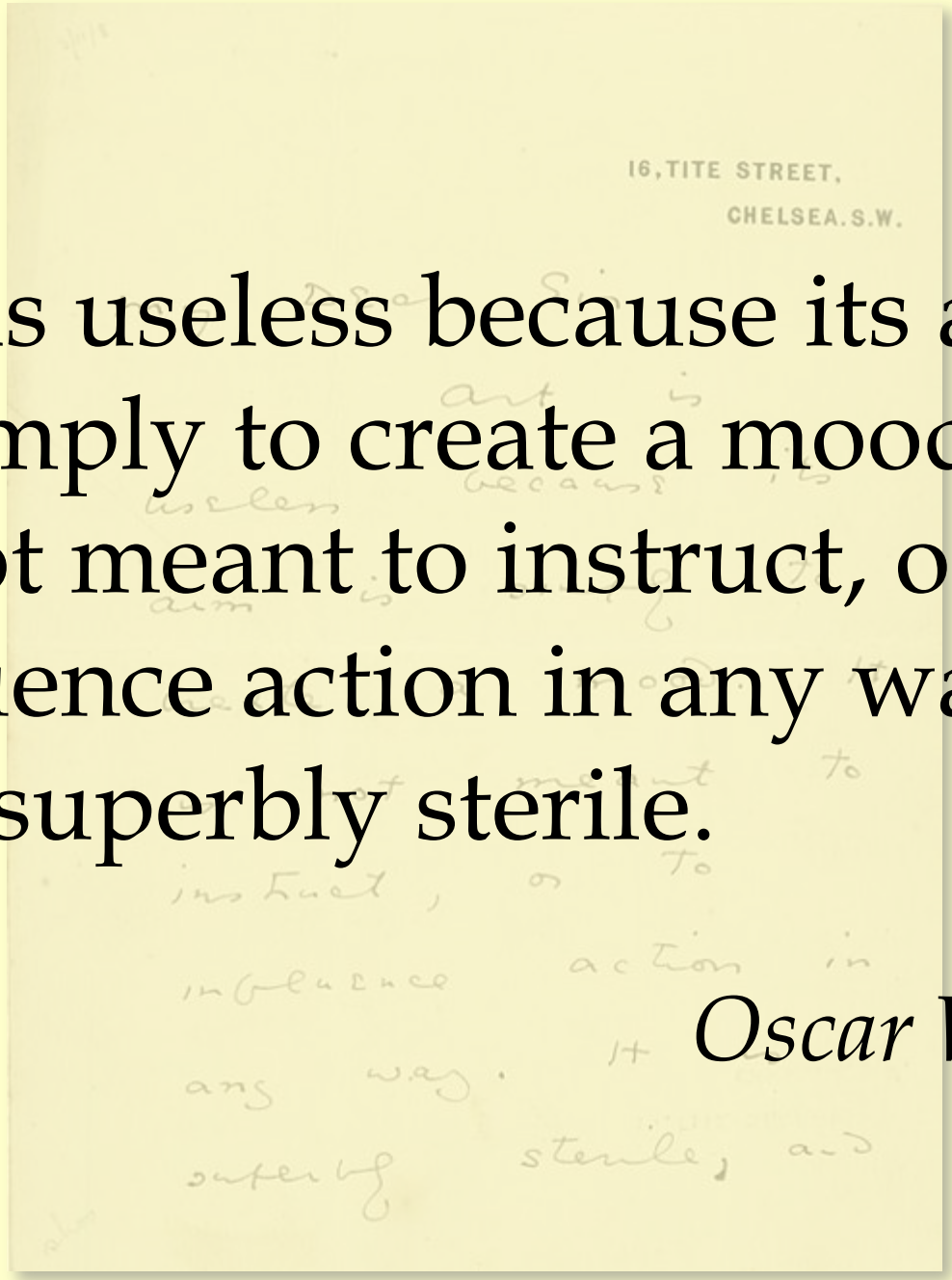
```
#!/bin/bash
function f() {
    sleep "$1"
    echo "$1"
}
while [ -n "$1" ]
do
    f "$1" &
    shift
done
wait
```


Useless
Machine
ON
OFF

16, TITE STREET,
CHELSEA. S.W.

My Dear Sir

Art is useless because its aim is simply to create a mood. It is not meant to instruct, or to influence action in any way. It is utterly sterile, and



Art is useless because its aim is simply to create a mood. It is not meant to instruct, or to influence action in any way. It is superbly sterile.

Oscar Wilde