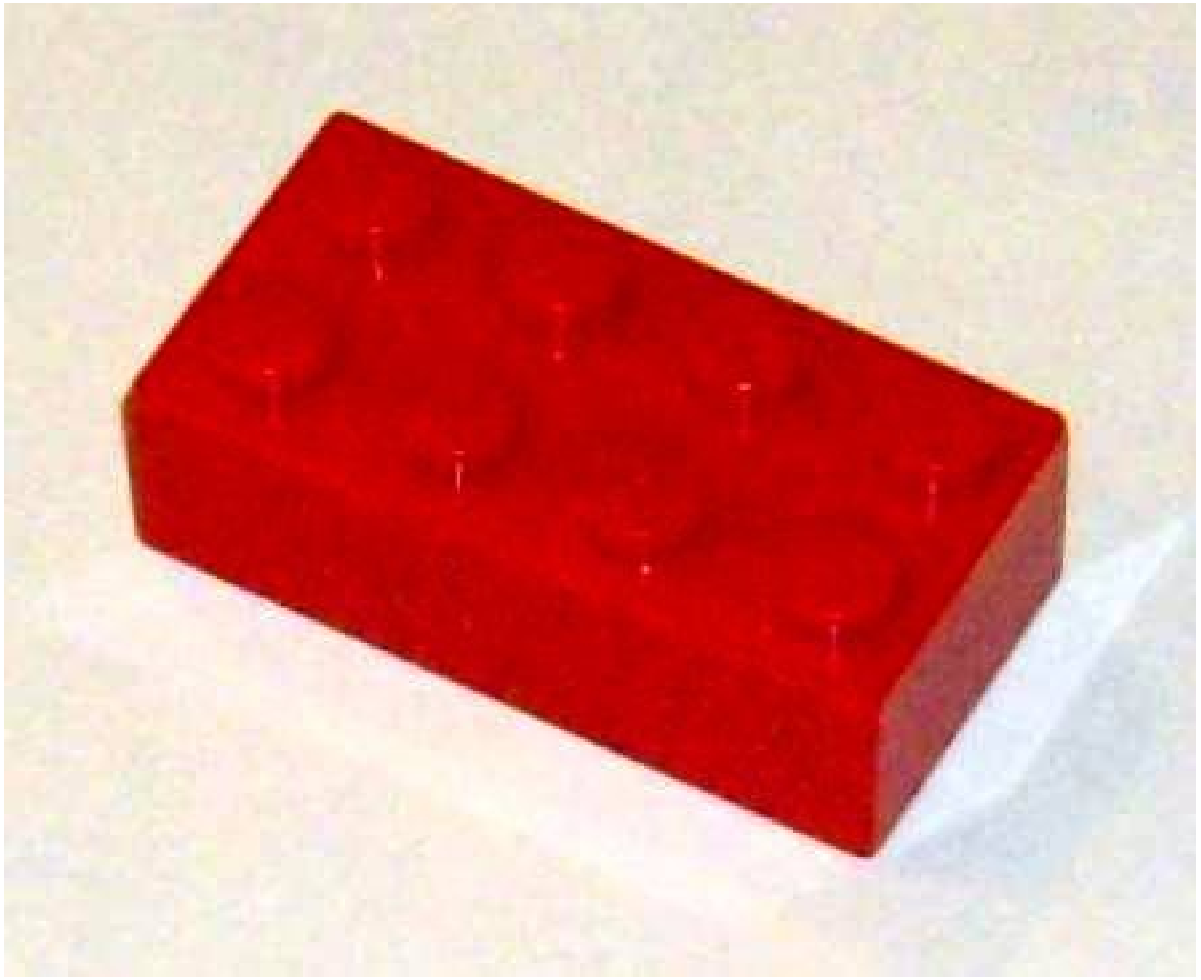
The D Programming Language

by Walter Bright
Digital Mars

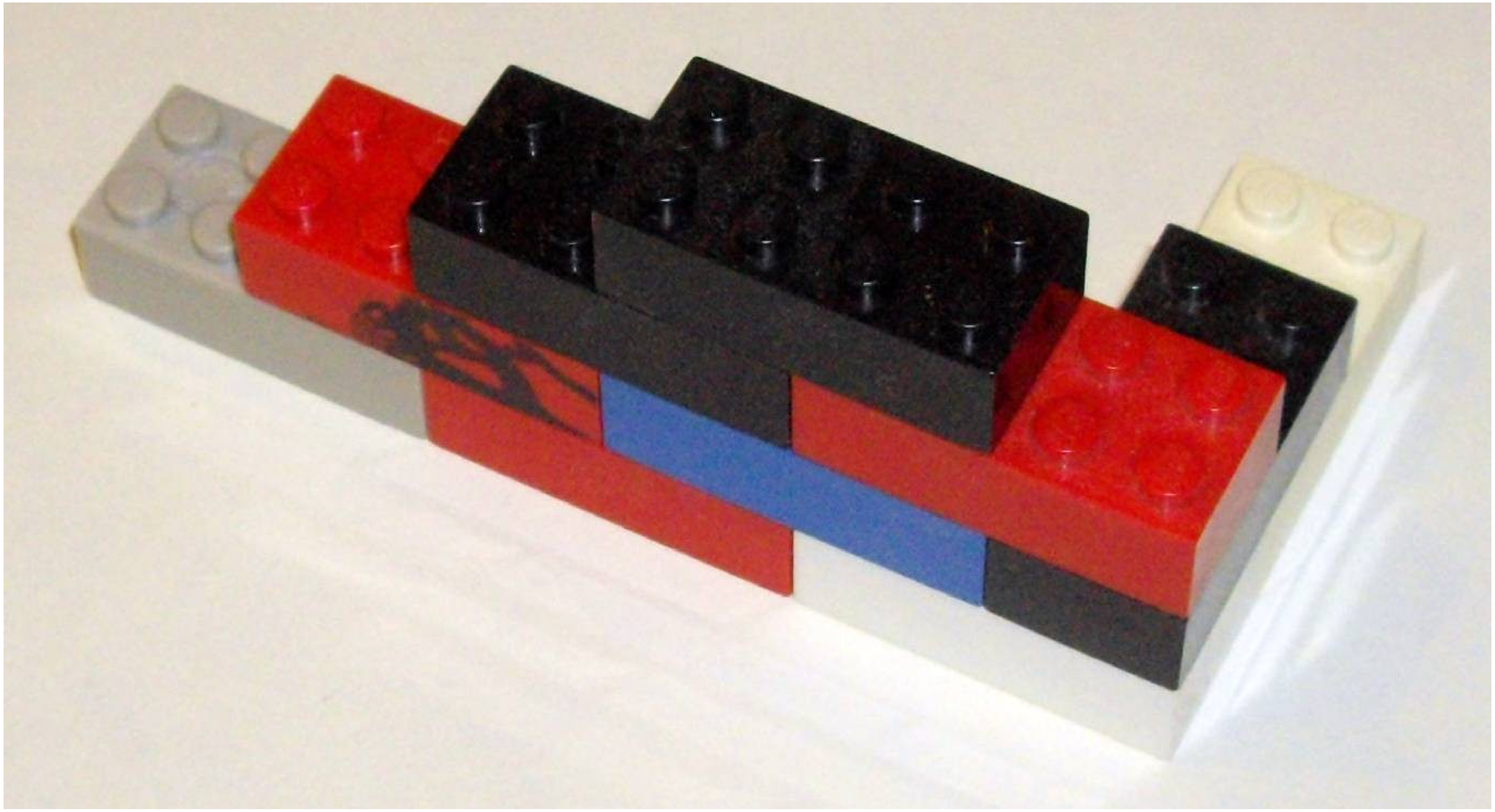
http://dlang.org/

Why?

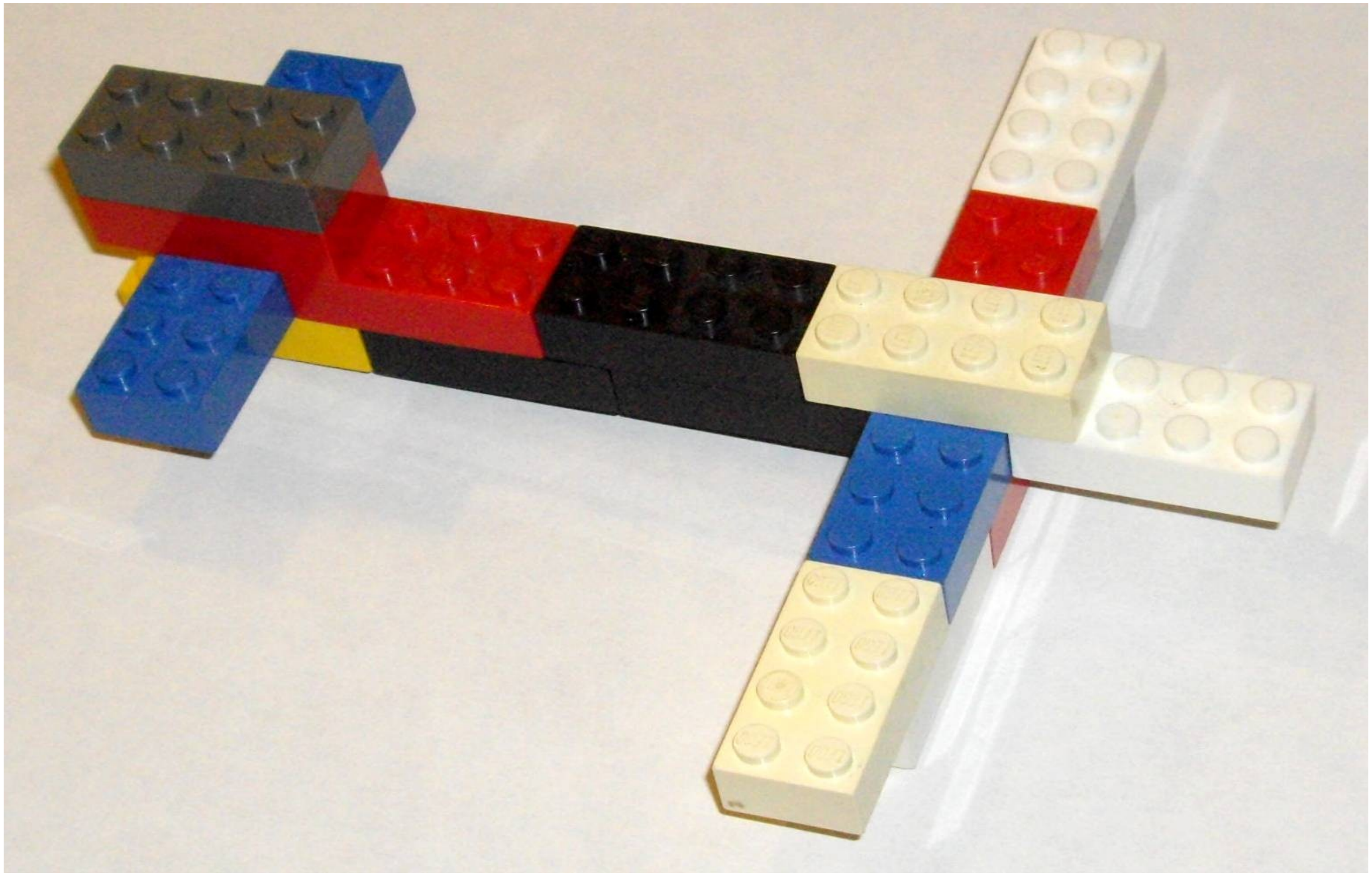
Modeling Power?



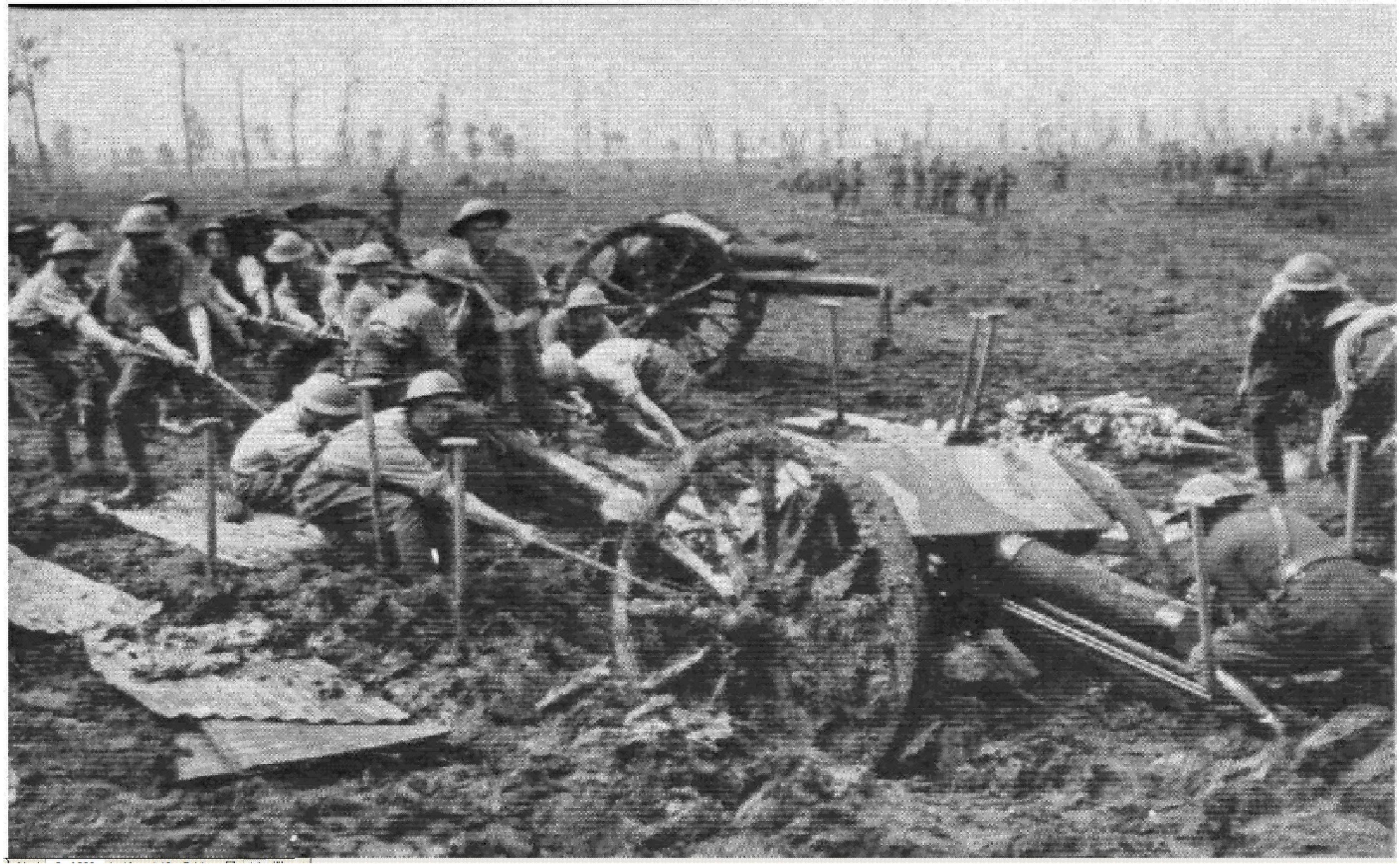
Great For Bricklike Models



Airplane?

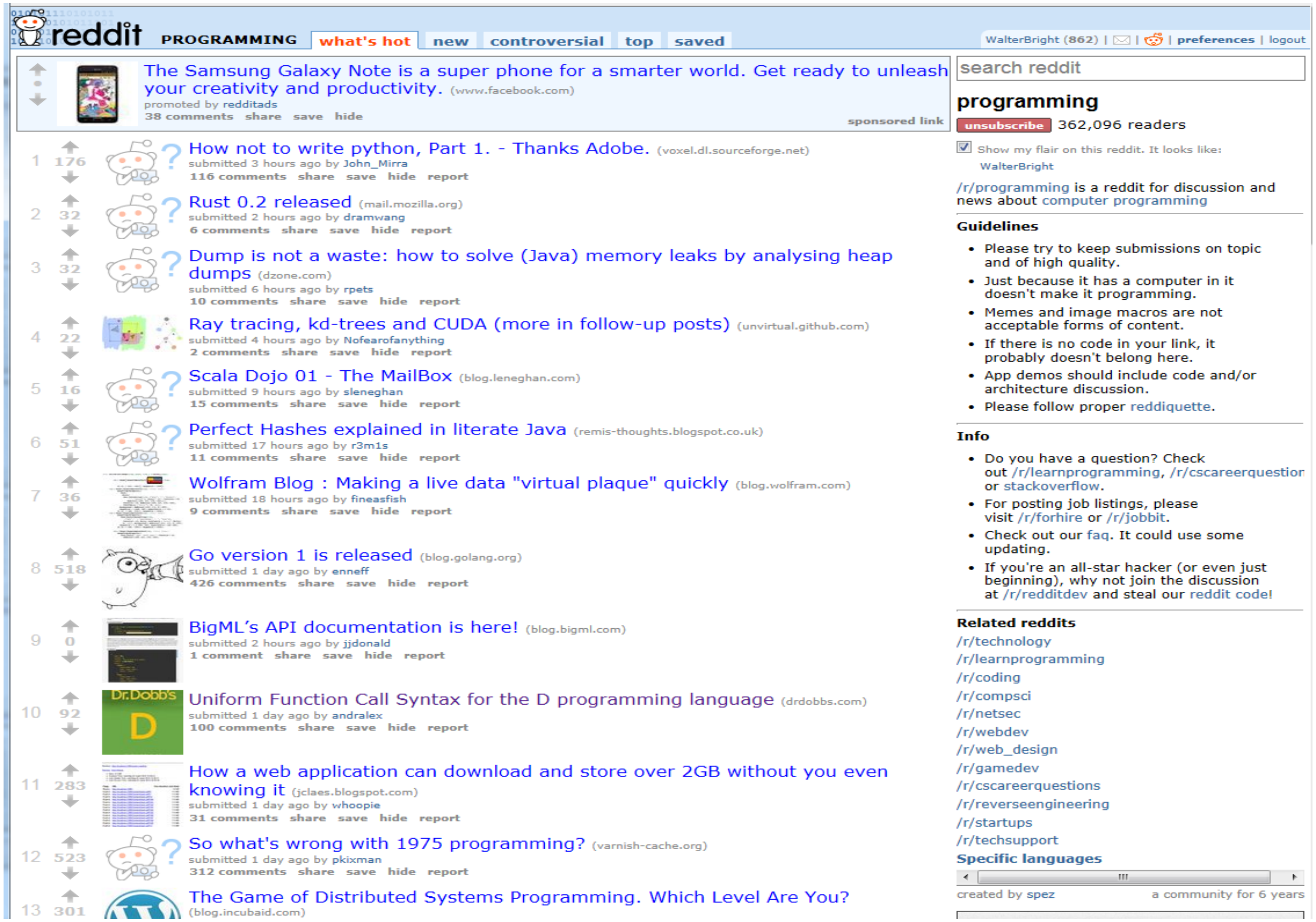


Obsolete Technology



Waiting?

Anything New On Reddit?



reddit PROGRAMMING | what's hot | new | controversial | top | saved

WalterBright (862) | | | preferences | logout

The Samsung Galaxy Note is a super phone for a smarter world. Get ready to unleash your creativity and productivity. (www.facebook.com)
promoted by redditads
38 comments share save hide
sponsored link

1 176 How not to write python, Part 1. - Thanks Adobe. (voxel.dl.sourceforge.net)
submitted 3 hours ago by John_Mirra
116 comments share save hide report

2 32 Rust 0.2 released (mail.mozilla.org)
submitted 2 hours ago by dramwang
6 comments share save hide report

3 32 Dump is not a waste: how to solve (Java) memory leaks by analysing heap dumps (dzone.com)
submitted 6 hours ago by rpets
10 comments share save hide report

4 22 Ray tracing, kd-trees and CUDA (more in follow-up posts) (unvirtual.github.com)
submitted 4 hours ago by Nofearofanything
2 comments share save hide report

5 16 Scala Dojo 01 - The MailBox (blog.leneghan.com)
submitted 9 hours ago by sleneghan
15 comments share save hide report

6 51 Perfect Hashes explained in literate Java (remis-thoughts.blogspot.co.uk)
submitted 17 hours ago by r3m1s
11 comments share save hide report

7 36 Wolfram Blog : Making a live data "virtual plaque" quickly (blog.wolfram.com)
submitted 18 hours ago by fineasfish
9 comments share save hide report

8 518 Go version 1 is released (blog.golang.org)
submitted 1 day ago by enneff
426 comments share save hide report

9 0 BigML's API documentation is here! (blog.bigml.com)
submitted 2 hours ago by jjdonald
1 comment share save hide report

10 92 Uniform Function Call Syntax for the D programming language (drdobbs.com)
submitted 1 day ago by andralex
100 comments share save hide report

11 283 How a web application can download and store over 2GB without you even knowing it (jclaes.blogspot.com)
submitted 1 day ago by whoopie
31 comments share save hide report

12 523 So what's wrong with 1975 programming? (varnish-cache.org)
submitted 1 day ago by pkixman
312 comments share save hide report

13 301 The Game of Distributed Systems Programming. Which Level Are You? (blog.incubaid.com)

search reddit

programming

unsubscribe 362,096 readers

☑ Show my flair on this reddit. It looks like:
WalterBright

/r/programming is a reddit for discussion and news about computer programming

Guidelines

- Please try to keep submissions on topic and of high quality.
- Just because it has a computer in it doesn't make it programming.
- Memes and image macros are not acceptable forms of content.
- If there is no code in your link, it probably doesn't belong here.
- App demos should include code and/or architecture discussion.
- Please follow proper reddiquette.

Info

- Do you have a question? Check out /r/learnprogramming, /r/cscareerquestior or stackoverflow.
- For posting job listings, please visit /r/forhire or /r/jobbit.
- Check out our faq. It could use some updating.
- If you're an all-star hacker (or even just beginning), why not join the discussion at /r/redditdev and steal our reddit code!

Related reddits

/r/technology
/r/learnprogramming
/r/coding
/r/compsci
/r/netsec
/r/webdev
/r/web_design
/r/gamedev
/r/cscareerquestions
/r/reverseengineering
/r/startups
/r/techsupport
Specific languages

created by spez a community for 6 years

Solutions exist in other languages,
but are not combined.

All In One Language:

Modeling Power

Modern Convenience

Native Efficiency

"You want to go forward, what do you do? You put it in D."

http://voices.washingtonpost.com/44/2010/08/obamas-latest-joke-republicans.html

D Has The Modeling Power

Mega Bloks P-51

The Right Modeling Paradigm for the Job

Polymorphism
Value semantics
Functional

Generics
Generative
Contract

Polymorphism

```
interface Shape {
  void Draw();
}
class Square : Shape {
  this(int xpos, int ypos, int width) {
    x = xpos; y = ypos;
    w = width;
  }
  void Draw() {
    writefln("Drawing Square at (%s,%s), width %s\n", x, y, width);
  }
  private int x, y, width;
}
```

Structs Have Value Semantics

```
struct BigNum {

    // construction
    this(int a) { … }

    // intercept copying
    this(this) { ... }

    // destructor
    ~this() { ... }
}
```

Functional

- Data immutability

```
immutable int[] a = [1, 2, 4, 6];
```

- Pure functions

```
pure int square(int x) { return x * x; }
```

- Lambda functions

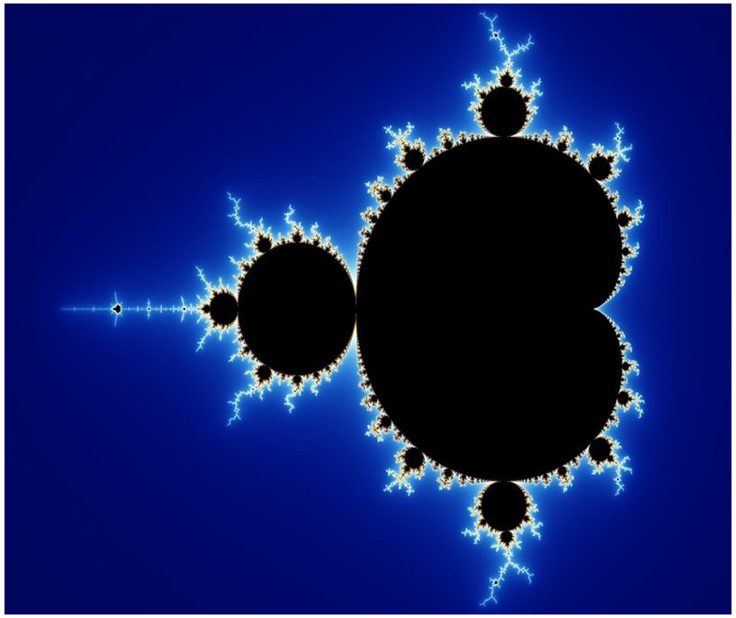
```
a.sort( (x,y) => x < y );
```

Generics

```
size_t levenshteinDistance
(alias equals = (a,b) => a == b, Range1, Range2)
(Range1 s, Range2 t)
if (isForwardRange!(Range1) &&
isForwardRange!(Range2))
{
    ...
}
```

Generative

```
struct A {
  int a;
  mixin(bitfields!(
    uint, "x", 2,
    int, "y", 3,
    uint, "z", 2,
    bool, "flag", 1));
}
A obj;
obj.x = 2;
obj.z = obj.x;
```



http://commons.wikimedia.org/wiki/File:Mandel_zoom_00_mandelbrot_set.jpg

Contract

```
// Interfaces and classes
interface Printable {
   void print(uint level)
   in { assert(level > 0); } // contract is part of the interface
}

// Interface implementation
class Widget : Printable {
   void print(uint level) { ... }
}

// Single inheritance of state
class ExtendedWidget : Widget {
   override void print(uint level)
   in { /* weakening precondition is okay */  }
   body {
       ... level may be 0 here ...
   }
}
```

Modern Convenience

Convenience

- Associative Arrays
- Static Typing With Inference
- Resource Management
- Slices and Ranges
- Immutability and Sharing is Typed
- System and Safe Code

Associative Arrays

```
// Symbol table
int[string] keywords = ["loop":3, "exit":4];
...
string abc;
...
if (keywords[abc] == 3)
  ...

// Sparse array of longs
long[int] sa;
sa[1] = 3;
sa[1000] = 16;
foreach (v; sa)
  writeln(v);
```

Static Typing With Inference

```
void main() {
    // Define an array of numbers, double[]. Compiler recognizes
    // common type of all initializers.
    auto arr = [ 1, 2, 3.14, 5.1, 6 ];
    // Dictionary that maps string to int, type is spelled int[string]
    auto dictionary = [ "one" : 1, "two" : 2, "three" : 3 ];
    // Calls the min function defined below
    auto x = min(arr[0], dictionary["two"]);
}

// Type deduction works for function results. Important for
// generic functions, such as min below, which works correctly
// for all comparable types.
auto min(T1, T2)(T1 lhs, T2 rhs) {
    return rhs < lhs ? rhs : lhs;
}
```

Resource Management

```
import std.stdio, core.stdc.stdlib;

class Widget { ... }

void main() {
    auto w = new Widget;  // automatic

    // Code is executed in any case upon scope exit
    scope(exit) { writeln("Exiting main."); }

    // RAII: File is closed deterministically at scope's end
    foreach (line; File("text.txt").byLine()) {
        writeln(line);
    }

    auto p = malloc(10);   // explicit C-style
    if (p) free(p);
}
```

Slices

```
auto filename = “etc/c/zip.d”;
auto path = filename[0..6];  // “etc/c/”
auto name = filename[6..9]; // “zip”
auto ext = filename[10..11]; // “d”
```

Safe, efficient, fast memory reuse

Ranges

```
#!/usr/bin/rdmd
import std.range, std.stdio;

// Compute average line length for stdin
void main() {
    ulong lines = 0, sumLength = 0;
    foreach (line; stdin.byLine()) {
        ++lines;
        sumLength += line.length;
    }
    writeln("Average line length: ",
        lines ? cast(double) sumLength / lines : 0.0);
}
```

Old Style

$$\int_{x_1}^{x_2} f(x)\, dx$$

```
double integrate(double x1, double x2,
  double delegate(double dx) f) {

  double sum = 0;
  double dx = 0.001;
  for (double x = x1; x < x2; x += dx)
    sum += f(x) * dx;
  return sum;
}

double parabola(double x, double c) {
  return integrate(0,x, (x => c * x * x));
}
```

Ranges and Algorithms

```d
import std.range : iota;
import std.algorithm : reduce;

auto integrate(alias f, T)(T x1, T x2) {
  auto dx = 0.001;
  return reduce!((a,x)=>a+f(x)*dx) (0.0, iota(x1, x2, dx));
}

auto parabola(double x, double c) {
  return integrate!(x => c * x * x)(0.0, x);
}
```

Immutability and Sharing is Typed

```
// Immutable data shared across threads
immutable string programName = "demo";

// Mutable data is thread-local
int perThread = 42;

// Explicitly shared data
shared int perApp = 5;
```

System and Safe Code

```
extern ( C ) @system void* calloc(size_t, size_t);

@trusted T[] callocArray(T)(size_t length) {
    T* p = cast(T*)calloc(length, T.sizeof);
    return p[0 .. length];
}

@safe int[] odds() {
    auto a = callocArray!int(3);
    a[0] = 1; a[1] = 3; a[2] = 5;
    return a;
}
```

Native Efficiency

Compiles to Native Code

- Fast program loads
- Predictable behavior
- Testing and QA is under your control
- You control the performance, not the VM vendor

Direct Access To C

```
void livingDangerously() {

    // Access to C's malloc and free primitives
    auto buf = malloc(1024 * 1024);
    scope(exit) free(buf); // free automatically upon scope exit

    // Interprets memory as an array of floats
    auto floats = cast(float[]) buf[0 .. 1024 * 1024];

    // Even stack allocation is possible
    auto moreBuf = alloca(4096 * 100);
    ...
}
```

Inline Assembler

```
uint checked_multiply(uint x, uint y) {
    uint result;
    version (D_InlineAsm_X86) {
        // Inline assembler "sees" D variables.
        asm {
            mov     EAX,x         ;
            mul     EAX,y         ;
            mov     result,EAX    ;
            jc      Loverflow     ;
        }
        return result;
    } else {
        result = x * y;
        if (!y || x <= uint.max / y)
           return result;
   }
Loverflow:
   throw new Exception("multiply overflow");
}
```

D Combines:

Modeling Power

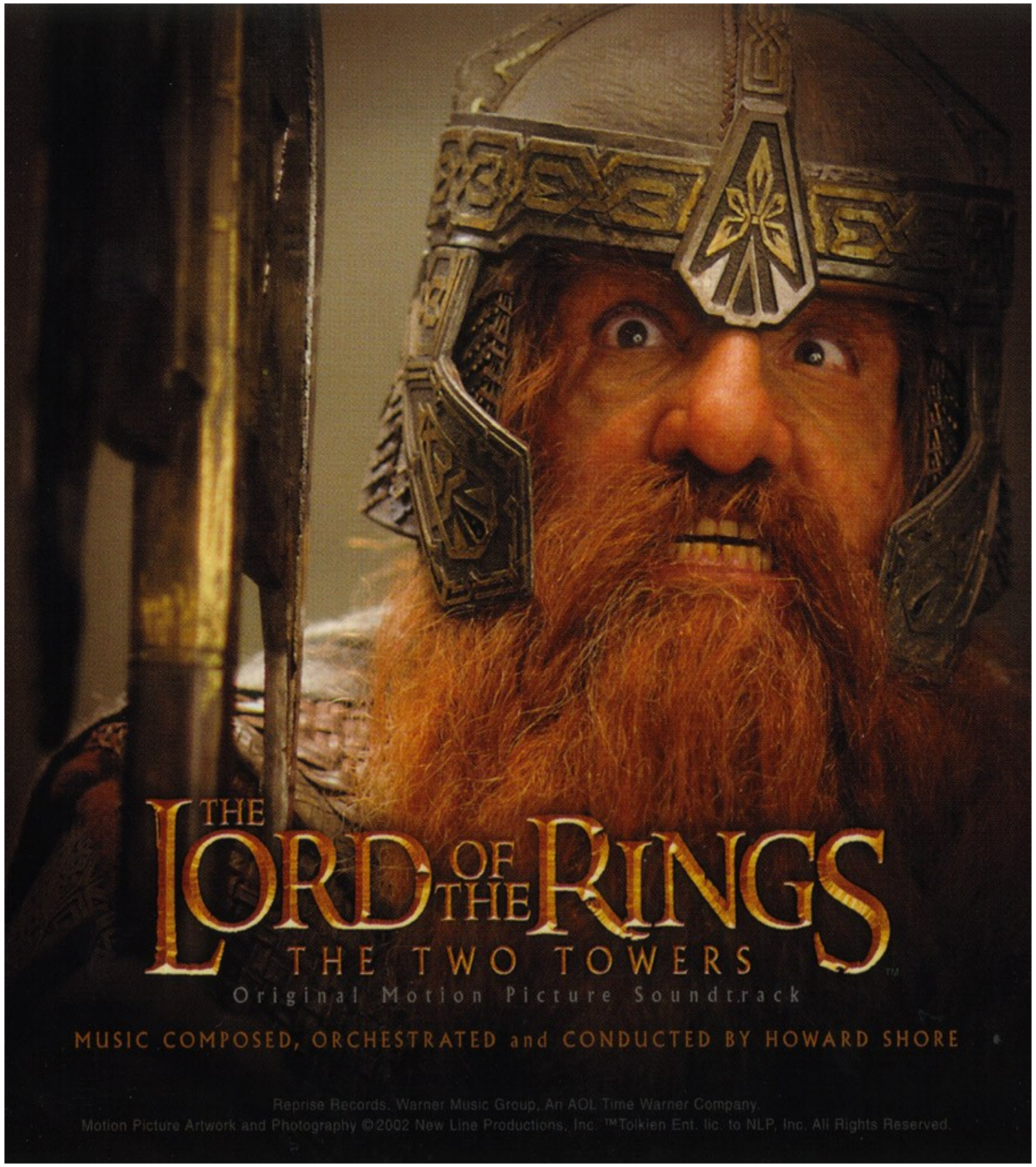
Modern Convenience

Native Efficiency

Can D go Mainstream?

Certainty of death

Very small chance of success


THE LORD OF THE RINGS
THE TWO TOWERS
Original Motion Picture Soundtrack
MUSIC COMPOSED, ORCHESTRATED and CONDUCTED BY HOWARD SHORE
Reprise Records, Warner Music Group, An AOL Time Warner Company.
Motion Picture Artwork and Photography ©2002 New Line Productions, Inc. ™Tolkien Ent. lic. to NLP, Inc. All Rights Reserved.

What Are You Waiting For?

Web site:

dlang.org

Community:

forum.dlang.org

Contribute:

https://www.github.com/D-Programming-Language

D Conference Sept. 26-29 2012

astoriaseminar.com