The D Programming Language

Lionello Lunesu – owtware.com
lio@lunesu.com @Lio李欧

Hello world

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

/// The entry point of the application.
/// Params:
///  args = the command line arguments
void main(string[] args)
{
    writeln("世界,你好!");
}
```

Introduction

- Modern convenience. Native efficiency.
- Systems programming language
- Community driven, open-source
- Multi-paradigm: IP, FP, OOP, DBC, TDD
- Multi-platform: Windows, OSX, Linux, FreeBSD
- Three popular toolsets: DMD, GDC, LDC

http://dlang.org/

Turtles all the way down

DConf 2013

- May 1st-3rd, Menlo Park, California
- Facebook, Sociomantic, Remedy, + Kickstarter
- 3 days, 19 speakers, ~70 attendees

DConf 2013 – Talks

- Distributed caching compiler
- Using D in games
- Concurrent garbage collector
- GDC: The GNU D Compiler
- Porting C# code to D
- Vibe.d: Web Development
- Precise garbage collector
- Higgs: Javascript JIT compiler
- AnalyzeD: Code analysis tool
- LDC: The LLVM D Compiler

DConf 2014

- May 21st - 23rd
- Menlo Park, Facebook HQ

http://dconf.org/

Productivity

High level vs. Low level

- Traditional trade-off:

	Build time	Run time	Productivity
Low level	-	+	-
High level	+	-	+
D	+	+	+

- D has best of both worlds

Bug-free software

D's Unique features

- Contracts and invariants
- Scope guards
- Immutability and purity
- Compile-time reflection
- Thread-local globals
- Compile-time function evaluation (CTFE)
- Uniform function call syntax (UFCS)
- mixin

Built-in types

- Signed integers: byte, short, int, long
- Unsigned integers: ubyte, ushort, uint, ulong
- Floating point: float, double, real
- Arrays: int[], float[]
- Strings: char[], wchar[], dchar[] (UTF encoded)
- Associative arrays: int[float]

Contracts

```
int FastDiv(int n, int d) {
in { assert(d != 0); }
out(r) { assert(r == n / d); }
body {
…
}
```

Invariant

```
class A {
  invariant() {
    assert(state > 0);
  }
  int state;
}
```

Scope guards

```
void main() {
   auto f = open("/dev/x", O_RDONLY);
   scope(exit) close(f);
   …
}
```

Immutability

```
//alias immutable(char)[] string;

void print(in char[] t) {  //const(char)[]
…
}

void main(string[] args) {
  foreach(a; args)
    print(a);
}
```

Compile-time reflection

```
T[] range(T)(uint count) pure
  if (__traits(compiles, T.init+1))
{
  T[] ar = new T[count];
  foreach(i, ref a; ar)
    a = i;
  return ar;
}
```

Thread-local globals

```
int g_ThreadLocal;
shared int g_Safe;
__gshared int g_Unsafe;

immutable A g_SafeA;
```

Compile-time function evaluation

```
pragma(msg, range!double(10));
```

```
$ dmd -c a.d
[0.00000, 1.00000, 2.00000, 3.00000,
4.00000, 5.00000, 6.00000, 7.00000,
8.00000, 9.00000]
```

Uniform Function Call Syntax

```
class A {…}

void someFunction(A instance) {…}

void main() {
   auto a = new A;
   a.someFunction();
}
```

mixin

```
string parseDSL(string text) pure {
  return `void validDcode() {}`;
}

mixin(parseDSL(`
…
`));
```

Resources

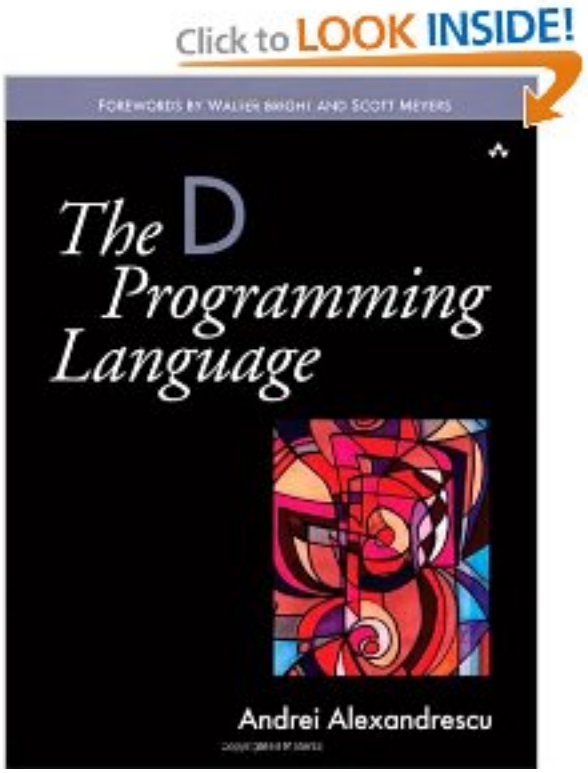
The D Programming Language

ISBN: 978-0321635365

http://noun.ly/study

http://dlang.org/

https://noun.ly/language



Workshops

- @新车间 XinCheJian

- Introduction to programming
- Advanced programming

Interested? Mailto: staff@xinchejian.com

Questions?