

# Современный Perl

Вячеслав Тихановский (vti)

2012-11-24

# 1. Инфраструктура Perl и краткая история развития языка



C, sh, awk, sed

18 декабря 1987 – 1.0

"Если есть задача, которая обычно решается с помощью `sed`, `awk` или `sh`, но превосходит их возможности или должна выполняться быстрее и не хочется писать все на `C`, то `perl`, возможно, вам подойдет."

1988 – 2.0: 1.0 + Улучшенные Regexp

1989 – 3.0: 2.0 + Бинарные потоки  
pack/unpack

1991 – 4.0: 3.0 + Документация



*There's More Than One Way To Do It*

**3rd Edition**  
Revised & Updated



*Programming*

Perl

**O'REILLY\***

*Larry Wall, Tom Christensen & Jon Orwant*

17 октября 1994 – 5.0: Модули

26 октября 1995 – СРАН

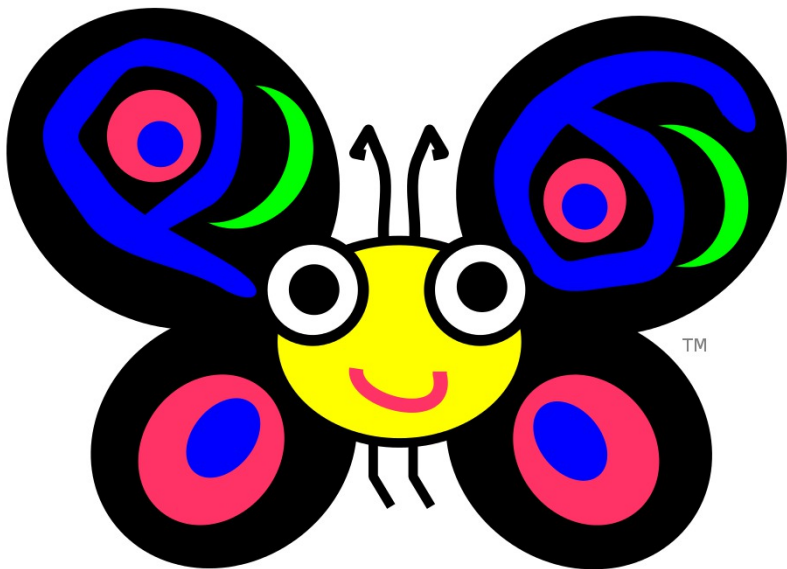
1997 – CGI.pm

1998 – "Изолента, которая держит интернет  
вместе"





2000 – Perl 6





2002 – Perl 5.8

Perl мертв?

2007 – Perl 5.10



2012 – Perl 5.16

Версии: нечетные и четные  
5.12 5.13

Релизы

Каждые три месяца в 20-х числах (pumpking)

## Perl 5.16 and beyond (Jesse Vincent)



- ▶ обратная совместимость
- ▶ use v5.14;
- ▶ все в модули!

Как один символ может изменить ваше  
Perl-резюме?

### 5.17.5 (37 авторов)

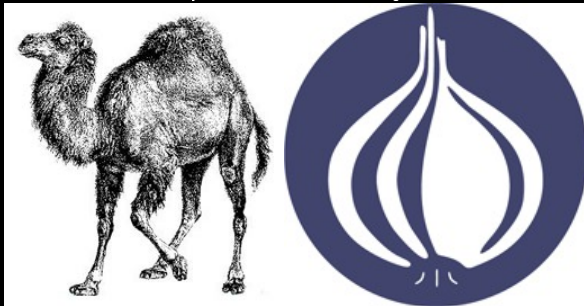
Abhijit Menon-Sen, Andy Dougherty, Aristotle Pagaltzis, Chris 'BinGOs' Williams, Colin Kuskie, Craig A. Berry, Daniel Dragan, David Golden, David Leadbeater, David Mitchell, David Nicol, Dominic Hargreaves, Father Chrysostomos, Florian Ragwitz, H.Merijn Brand, James E Keenan, Jerry D. Hedden, Jesse Luehrs, John Peacock, Karen Etheridge, Karl Williamson, Leon Timmermans, Michael G Schwern, Nicholas Clark, Peter Martini, Rafael Garcia-Suarez, Ricardo Signes, Shawn M Moore, Shlomi Fish, Steffen Müller, Steve Hay, Sullivan Beck, Sébastien Aperghis-Tramoni, Tony Cook, Vincent Pit, Yves Orton.

AUTHORS (1151)

Perl, perl или PERL?

Only perl can parse Perl.

Верблюд или лук?



# Философия Perl

TIMTOWTDI

There Is More Than One Way To Do It



## Как очистить массив?

```
1 @a = ();  
2 undef @a;  
3 splice @a;  
4 $#a = -1;  
5 delete @a[0..#a];  
6 shift @a while defined shift @a;  
7 while (shift @a) {}
```

Мультипарадигменность  
Процедурный Функциональный  
Объектно-ориентированный  
Аспектно-ориентированный

DWIM  
Do What I Mean

Три добродетели программиста:  
лень, нетерпение и высокомерие

# Доступность и открытость

# Artistic и GPL

## Более 100 платформ

Acorn AIX Amiga Apollo Apple Atari AtheOS BeOS  
BSD BSD/OS Coherent Compaq Concurrent Cygwin  
Darwin DEC OSF/1 DG/UX Digital Digital UNIX  
DJGPP DOS Domain/OS DragonFlyBSD  
DYNIX/ptx Embedix EMC EPOC FreeBSD Fujitsu  
GNU Darwin Guardian HP HP-UX IBM IRIX  
Japanese JPerl Linux LynxOS Macintosh Mac OS X  
Mac OS MinGW Minix MiNT MorphOS MPE/iX  
MS-DOS / MVS NetBSD NetWare NextStep  
NonStop NonStop-UX Novell OpenBSD OpenVMS  
OS/2 OS/390 OS/400 OSF/1 Plan 9 Pocket PC  
PowerMAX Psion QNX Reliant UNIX RISCOS  
Sequent SGI Sharp Siemens SINIX Solaris Stratus  
Sun Syllable Symbian Tandem Tivo Tru64 UNIX ...

Сборки для Windows  
ActiveState StrawberryPerl Cygwin CitrusPerl  
DWIMPerl

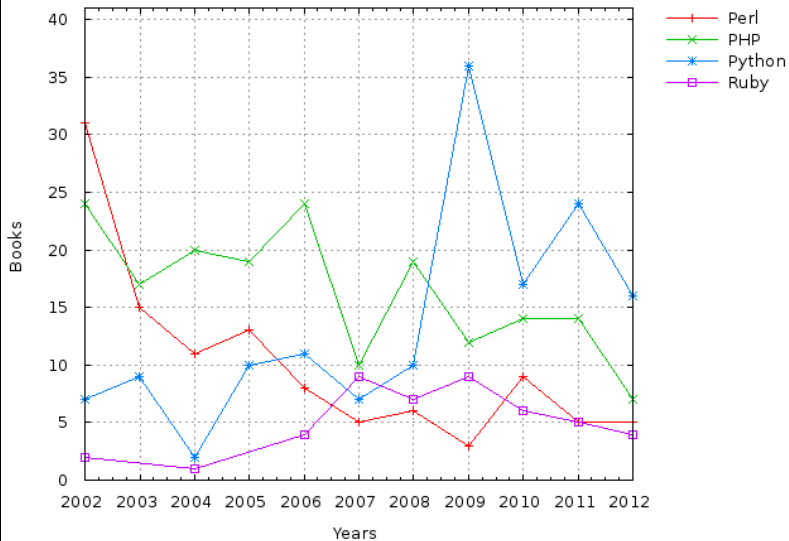


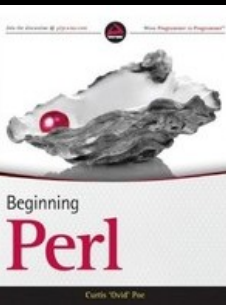
Свежая документация

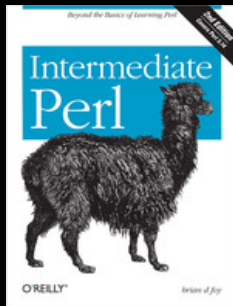
Perl-документация [perldoc.perl.org](http://perldoc.perl.org)  
Perl-уроки [perl-tutorial.org](http://perl-tutorial.org), [perltuts.com](http://perltuts.com)

## Свежие книги

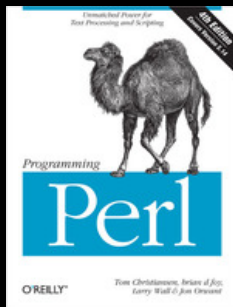
Books published per year







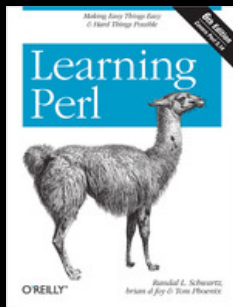
2 издание



4 издание







6 издание



# H I G H E R O R D E R P E R L



TRANSFORMING PROGRAMS WITH PROGRAMS

PRECI

MARK JASON DOMINUS

# CPAN

## Comprehensive Perl Archive Network

CTAN (T<sub>E</sub>X) -> CPAN -> JSAN (JavaScript) ->  
CCAN (C)  
pypi (Python) Gems (Ruby) npm (NodeJS)

CPAN-статистика

- > 115k модулей
- > 26k дистрибутивов
- > 10k авторов

Поиск по СРАН



<a href="#">Archiving/Compression/Conversion</a>	<a href="#">File Name Systems/Locking</a>	<a href="#">Option/Parameter/Config/Processing</a>
<a href="#">Bundles (and SDKs)</a>	<a href="#">Graphics</a>	<a href="#">Perl6</a>
<a href="#">Commercial Software Interfaces</a>	<a href="#">Internationalization/Locale</a>	<a href="#">Pragmas</a>
<a href="#">Control Flow Utilities</a>	<a href="#">Language Extensions</a>	<a href="#">Security</a>
<a href="#">Data and Data Types</a>	<a href="#">Language Interfaces</a>	<a href="#">Server/Daemon Utilities</a>
<a href="#">Database Interfaces</a>	<a href="#">Mail and Usenet News</a>	<a href="#">String/Language/Text Processing</a>
<a href="#">Development Support</a>	<a href="#">Miscellaneous</a>	<a href="#">User Interfaces</a>
<a href="#">Documentation</a>	<a href="#">Networking Devices/IPC</a>	<a href="#">World Wide Web</a>
<a href="#">File Handle Input/Output</a>	<a href="#">Operating System Interfaces</a>	

86246 Upgrades, 28197 Distributions  
155077 Modules, 10190 Upgrades

hosted by [Zalando](#)



# meta::cpan

Search the CPAN

fm Feeling Lucky

Hint: Press shift and enter if you are feeling lucky

[API](#)

[About MetaCPAN](#)

[Fork metacpan.org](#)

Hosting generously  
sponsored by  speedchilli  
per arch partner



API

## My Perl Résumé

---

Your Perl Résumé is generated automatically based on your [CPAN](#) experience, activity and impact.

PAUSE ID

Generate

### See some examples

[Christian Walde](#)  
[David Precious](#)  
[Marc Lambrichs](#)  
[Dominique Dumont](#)  
[Diego Kuperman](#)  
[David E. Wheeler](#)  
[Logan Bell](#)  
[Bogdan Lucaciu](#)  
[Kent Fredric](#)  
[Stanislaw Pusep](#)

### Notes

All the data is fetched from the awesome [MetaCPAN](#).

[Show comments](#)

---

# perlybook.org

[CPAN Perlbits About Blog](#) *Release v0.15*

## PerlyBook

☒ fetch complete release

EPUB

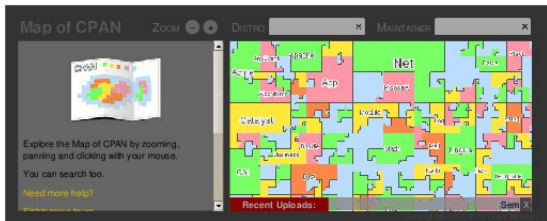
MOBI

POD: Pod On Demand.


This website just came in to the world. Please report any problems [here](#).



# mapofcpan.org



# Тестирование CPAN – CPANTS



## CPAN Testers Reports

www.cpan testers .org  
Reports for Protocol-WebSocket

Blog | Reports | Wiki | Statistics | PASS Matrix | Preferences | Database | Development | CPAN Dependencies | CPAN Matrix | Analysis

> Home > Distributions > P > Protocol-WebSocket

### Reports for Protocol-WebSocket

This page was last updated on **Tuesday, 20th November 2012 (8:15pm)**. [JSON](#) [RSS](#) [YAML](#)

#### PASS Summary

The following table lists the most recent version of Protocol-WebSocket to have had at least one PASS report submitted for the given perl version and platform. Note that we list the generic platform name, and reports may have been processed for different release versions of that operating system.

Perl Version	Windows (Cygwin)	Mac OS X	Dragonfly BSD	FreeBSD	Debian GNU/Linux BSD	Linux	GNU/Linux	NetBSD	Windows (Win32)
5.17.6							0.12		
5.17.5							0.11		
5.17.4							0.12		
5.17.3							0.12	0.11	
5.17.2							0.12		
5.17.1							0.12		
5.17.0							0.09006		
5.16.2							0.12		
5.16.1		0.12		0.11			0.12	0.11	0.12

Menu

- Home
- Status
- Help
- About
- Admin
- Static Website
- Find A Tester

Preferences

Grade:  
All Reports

Availability:  
On CPAN ONLY

Dist Maturity  
Official ONLY

Perl Maturity  
Official ONLY

Patched Perl  
Exclude Patches

Perl Version  
ALL

OS Name  
ALL

# CPAN рейтинг

 Ratings and Reviews for CPAN

cpanratings



HOME SEARCH ABOUT SITECARD ACCOUNT LOGIN

## Rate a module

Find a distribution you would like to review:

## Recent reviews [\(RSS feed\)](#)

### [Text-WordGrams](#) (0.06 rc)

"Calculates statistics on word ngrams." Uh... statistics? Where? This module just parses into ngrams and does no stats.

Check out Text::Ngrams.

[Gene Boggs](#) - 2012-11-19T09:45:36 ([permalink](#))

Was this review helpful to you? [Yes](#) [No](#)

### [Promises](#) (0.01 3) ★★★★★

Truly well-written documentation. Just reading the pod was an education.

I don't do much async programming in Perl, but next time I do, I will "use Promises".

[Toby Inster](#) - 2012-11-19T05:44:20 ([permalink](#))

Was this review helpful to you? [Yes](#) [No](#)

### [HTML-TreeBuilder-XPath](#) (0.14 rc) ★★★★★

Thanks for this, I'm coding many web-scrapers for years, and HTML-TreeBuilder-XPath is quite easier & quicker to write code than HTML-TreeBuilder alone.

[Giles Querol](#) - 2012-11-18T09:54:30 ([permalink](#))

Was this review helpful to you? [Yes](#) [No](#)

1 out of 1 found this review helpful:

# CPAN баги

RT for rt.cpan.org

## Bug #80160 for Mojo-JSON-Any: Won't pass Mojo::JSON's tests.

### Report information

#### The Basics

ID: 80160  
Status: new  
Priority: 0  
Queue: Mojo-JSON-Any

#### Bug Information

BrokenIn: (no value)  
FixedIn: (no value)  
Severity: Wishlist

#### People

Owner: Nobody in particular  
Requester: davido [...].cpan.org  
Cc:  
AdminCc:

#### Attachments

json.t

• Fri Oct 12 23:38:09 2012 (11.9K) by davido [...].cpan.org

### History

# Fri Oct 12 23:38:09 2012 davido [...].cpan.org - Ticket created  
Subject: Won't pass Mojo::JSON's tests.

As a near drop-in replacement for Mojo::JSON, Mojo::JSON::Any (with Mojo::JSON::XS) should be able to pass the Mojo::JSON test suite. Currently it fails tests #17, 19, 68, 72, 73, and 81. Each of those seem to be related to the fact that Mojo::JSON.t is expecting "1" (a string) whereas Mojo::JSON::XS is returning 1 (an integer). Tests 86 and 87 are also failing "properly escaped" tests.

Attached is a copy of Mojo::JSON.t, modified to instantiate and test Mojo::JSON::Any instead of Mojo::JSON. Whether or not Mojo::JSON::Any passes (or fails) Mojo::JSON's tests should be a strong indicator of whether or not regressions are introduced by the XS version.

Subject: json.t

Message body not shown because it is too large.

Вопросы?



## 2. Современные Best Practices

# Современная сборка Perl

```
perlbrew install 5.14.0  
perlbrew install perl-5.8.1  
perlbrew install perl-5.13.6  
  
perlbrew use perl-5.13.6
```

# Современная установка модулей

- ▶ cpan
- ▶ cpanp
- ▶ **cpanm** (local::lib)

# Развертывание

carton

```
> cat Makefile.PL
use inc::Module::Install;
name 'MyApp';
version '1.0';

requires 'Plack', 0.9980;
requires 'Starman', 0.2000;
```

```
WriteAll;
```

```
> carton install
> git add Makefile.PL carton.lock
> git commit -m "add Plack and Starman"

> carton install
```



# Best Practices

*Standards and Styles for Developing Maintainable Code*



# Perl Best Practices

O'REILLY®

*Dennis Gossage*

```
use strict;  
use warnings;
```

```
use ModernPerl;  
# or  
use common::sense;  
# or  
...
```

```
open FH, ">$filename";
```

```
open FH, '>', $filename;
```

```
open my $fh, '>', $filename;
```

```
new CGI;
```

```
CGI->new;
```

```
Plack::Request->new;
```

**use** v5 .xx .xx\* ;

\* .xx.xx > 8.8!!!

## Секретные операторы (<https://github.com/book/perlsecret/>)

0+, @[], !!, }{, -, - , - + -, = () =, =<>= ,  
<>, << *m* = *m* >> *m*;;, - =!- =!!, + =!+ =!!,  
x =!x =!!, \* =!\* =!!, , =>, ()x!!



say

**print** "Hi\n"

say "Hi"

//

\$foo ||= \$bar;

\$foo = \$bar **unless** **defined** \$foo;  
\$foo //=\$bar;

state

```
my $counter = 0;  
sub inc { $counter++ }  
  
sub inc {  
    state $counter = 0;  
    $counter++  
}
```

push/pop, shift/unshift

```
push $array_ref , 'bar';  
unshift $array
```

```
$foo ~~ undef  
$foo ~~ @array  
$foo ~~ %hash
```

```
$phone =~ m/^(.*)\.\/;  
print $1
```

```
my $PREFIX = qr/^(\d+)\./;  
my $NUMBER = qr/\\s*(\\d+)\\)\\s*(\\d+)$/;  
...  
$phone =~ m/$PREFIX$NUMBER/;  
print $1;  
print $2;  
print $3;
```

**\$** + *name*

```
my $PREFIX = qr/^(<prefix>d+)\./;  
my $NUMBER = qr/\s*\(<area>\d+)\)\s*  
...  
$phone =~ m/$PREFIX$NUMBER/;  
print $+{prefix};  
print $+{area};  
print $+{number};
```



## Regex::Common

[illegible]

SQL

DBIx::Class, Rose::DB::Object, ObjectDB

ООП

Moose, Moos, Moo, Mo, M  
Mouse, Mousse

```
package Foo;  
use Moose;
```

```
has 'name' => (  
    isa      => 'Str',  
    required => 1  
);
```

```
package main;
```

```
my $foo = Foo->new(name => 'bar');
```

# Method::Signatures



```
package Foo;  
use Method::Signatures;  
  
method get($key) {  
    return $self->{$key}  
}
```

# Веб-программирование

CGI, FCGI, mod\_perl  
PSGI/Plack

```
sub {  
  my $env = shift ;  
  
  return [200 ,  
          [ 'Content-Type' => 'text/plain' ],  
          [ 'Hello␣world!' ]  
        ]  
}
```

HTTP::Server::PSGI, HTTP::Server::Simple::PSGI,  
Starman, Twiggy, Corona, Feersum, Starlet,  
FCGI::EV, AnyEvent::FCGI, Danga::Socket,  
AnyEvent::HTTPD, SCGI, AnyEvent::SCGI, POE,  
ReverseHTTP, uWSGI, mod\_psgi, evpsgi, Perlbal,  
nginx embedded perl

Catalyst, Jifty, CGI::Application, HTTP::Engine,  
Dancer, Mason, Squatting, Continuity, Maypole,  
Tatsumaki, Mojolicious, Web::Simple, Angelos, Ark,  
Schenker, Noe, Kamui and WebNano

# Template::Toolkit

## Sass, Haml, Mustache

# HTML5, WebSockets

## Protocol::WebSocket, PocketIO, SockJS



# Тестирование и профилирование

Test::\*

Test::Builder, Test::Unit...  
Test::Pod, Test::NoWarnings, Test::TCP...

TDD/BDD  
Test::Class, Test::BDD::\*

```
use Test::lka;
use Test::Expect;

describe 'MessageFilter' => sub {
  my $filter;

  before_each {
    $filter = MessageFilter->new();
  };

  it 'should detect message with NG word' => sub {
    my $filter = MessageFilter->new('foo');
    expect($filter->detect('hello foo'))->ok;
  };

  it 'should detect message with NG word' => sub {
    my $filter = MessageFilter->new('foo');
    expect($filter->detect('hello foo'))->ok;
  };
};

runtests;
```

Devel :: Cover

cover --test

# Coverage Summary

Database:	/home/vti/dev/protocol-websocket/cover_db			
Report Date:	2012-11-22 14:24:42			
Perl Version:	v5.14.2			
OS:	linux			
Thresholds:	< 75%	< 90%	< 100%	= 100%

file	stmt	bran	cond	sub	pod	time	total
<a href="#">blib/lib/Protocol/WebSocket/Cookie.pm</a>	100.0	91.7	100.0	100.0	100.0	4.6	98.4
<a href="#">blib/lib/Protocol/WebSocket/Cookie/Request.pm</a>	100.0	77.3	n/a	100.0	100.0	0.2	93.0
<a href="#">blib/lib/Protocol/WebSocket/Cookie/Response.pm</a>	92.9	81.2	n/a	80.0	100.0	0.5	88.2
<a href="#">blib/lib/Protocol/WebSocket/Frame.pm</a>	97.0	85.7	61.3	95.8	62.5	9.3	88.8
<a href="#">blib/lib/Protocol/WebSocket/Handshake.pm</a>	95.2	75.0	66.7	88.9	100.0	0.4	88.9
<a href="#">blib/lib/Protocol/WebSocket/Handshake/Client.pm</a>	92.5	77.3	50.0	92.3	83.3	13.7	86.0
<a href="#">blib/lib/Protocol/WebSocket/Handshake/Server.pm</a>	88.5	93.8	100.0	81.8	100.0	13.6	90.4
<a href="#">blib/lib/Protocol/WebSocket/Message.pm</a>	98.1	87.1	70.0	100.0	100.0	8.0	92.8
<a href="#">blib/lib/Protocol/WebSocket/Request.pm</a>	96.2	89.4	73.9	100.0	91.7	25.4	91.6
<a href="#">blib/lib/Protocol/WebSocket/Response.pm</a>	97.7	81.2	90.9	100.0	100.0	18.7	93.5
<a href="#">blib/lib/Protocol/WebSocket/Stateful.pm</a>	68.8	50.0	n/a	87.5	100.0	1.8	76.5
<a href="#">blib/lib/Protocol/WebSocket/URL.pm</a>	100.0	87.5	69.2	100.0	100.0	3.8	92.4
Total	96.0	85.3	72.4	95.9	91.8	100.0	91.0

Devel :: NYTProf



# Performance Profile Index

For benchmark.pl

Profile of benchmark.pl for 526ms (of 553ms), executing 8620 statements and 3797 subroutine calls in 2

## Top 15 Subroutines

Calls	P	F	Exclusive Time	Inclusive Time	Subroutine
10	1	1	203ms	203ms	Protocol::WebSocket::Frame::_mask
10	1	1	156ms	156ms	Protocol::WebSocket::Frame::new
10	1	1	33.0ms	240ms	Protocol::WebSocket::Frame::to_bytes
10	1	1	28.2ms	424ms	main: __ANON__[benchmark.pl:22]
1544	3	1	23.8ms	28.9ms	Benchmark::new
2	2	1	11.0ms	465ms	Benchmark::runloop
1544	1	1	5.11ms	5.11ms	Benchmark::mytime
1	1	1	4.04ms	13.9ms	main: BEGIN@10
1	1	1	3.60ms	11.0ms	Protocol::WebSocket::Frame::BEGIN@7
1	1	1	2.93ms	3.42ms	Benchmark::BEGIN@432
1	1	1	2.81ms	3.03ms	Config::AUTOLOAD
1	1	1	2.61ms	5.78ms	Benchmark::BEGIN@453
1	1	1	2.45ms	2.49ms	main: BEGIN@4
1	1	1	2.17ms	16.9ms	main: BEGIN@8
1	1	1	1.96ms	3.74ms	Encode::BEGIN@48

See [all 324 subroutines](#)

# Source Code Files — ordered by exclusive time then name

Stmts	Exclusive Time	Reports	Source File
497	395ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	lib/Protocol/WebSocket/Frame.pm (including 1 string eval)
6428	45.1ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Benchmark.pm (including 2 string evals)
31	31.6ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	benchmark.pl
78	3.59ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Encode.pm
369	3.40ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Encode/Alias.pm
48	2.98ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Carp.pm
43	2.85ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Config_heavy.pl
106	2.51ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Exporter/Heavy.pm
66	2.22ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	DynaLoader.pm
70	2.07ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	base.pm (including 2 string evals)
116	1.42ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	vars.pm
72	1.36ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	constant.pm
81	1.28ms	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Config.pm
237	944µs	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	warnings.pm
11	941µs	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Encode/Config.pm
31	763µs	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	lib.pm
21	741µs	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Time/HiRes.pm (including 1 string eval)
169	730µs	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	strict.pm
92	633µs	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Exporter.pm
13	587µs	<a href="#">line</a> • <a href="#">block</a> • <a href="#">sub</a>	Scalar/Util.pm (including 1 string eval)

297					
298					# spent 203ms within Pro # 10 times (203ms+0s) by sub _mask {
299	10	16μs			my \$self = shift;
300	10	24.4ms			my (\$payload, \$mask)
301					
302	10	39.5ms			\$mask = \$mask x (int
303	10	44.3ms			\$mask = substr(\$mask,
304	10	28.6ms			\$payload ^= \$mask;
305					
306	10	66.4ms			return \$payload;
307					}
308					
309	1	6μs			1;
310					__END__

write-only

perltidy

```
1 sub new { my $class = shift; my $self = {captures => [],
2   constraints => {}, @_}; bless $self, $class; if (my
3   $methods = $self->{method}) { $methods = [$methods]
4   unless ref $methods eq 'ARRAY'; $methods = [map {uc}
5     @$methods]; $self->{method} = $methods; }
6   $self->_prepare_pattern; return $self; }
```

```
1  sub new {
2      my $class = shift;
3      my $self = { captures => [], constraints => {}, @_ };
4      bless $self, $class;
5      if ( my $methods = $self->{method} ) {
6          $methods = [$methods] unless ref $methods eq 'ARRAY';
7          $methods = [ map { uc } @$methods ];
8          $self->{method} = $methods;
9      }
10     $self->_prepare_pattern;
11     return $self;
12 }
```

# Perl::Critic



Always unpack @\_ first at line 13, column 1. See page 178 of PBP. (Severity: 4)  
Subroutine "cookies" does not end with "return" at line 80, column 1. See page 197 of PBP.  
Always unpack @\_ first at line 82, column 1. See page 178 of PBP. (Severity: 4)  
Subroutine "resource\_name" does not end with "return" at line 82, column 1. See page 197 of PBP.  
Subroutine "upgrade" does not end with "return" at line 86, column 1. See page 197 of PBP.  
Subroutine "connection" does not end with "return" at line 87, column 1. See page 197 of PBP.  
Always unpack @\_ first at line 89, column 1. See page 178 of PBP. (Severity: 4)  
Subroutine "number1" does not end with "return" at line 89, column 1. See page 197 of PBP.  
Always unpack @\_ first at line 90, column 1. See page 178 of PBP. (Severity: 4)  
Subroutine "number2" does not end with "return" at line 90, column 1. See page 197 of PBP.  
Always unpack @\_ first at line 92, column 1. See page 178 of PBP. (Severity: 4)  
Subroutine "key" does not end with "return" at line 92, column 1. See page 197 of PBP.  
Always unpack @\_ first at line 93, column 1. See page 178 of PBP. (Severity: 4)  
Subroutine "key1" does not end with "return" at line 93, column 1. See page 197 of PBP.  
Always unpack @\_ first at line 94, column 1. See page 178 of PBP. (Severity: 4)  
Subroutine "key2" does not end with "return" at line 94, column 1. See page 197 of PBP.  
Regular expression without "/x" flag at line 114, column 16. See page 236 of PBP.  
Regular expression without "/x" flag at line 132, column 21. See page 236 of PBP.  
Always unpack @\_ first at line 174, column 1. See page 178 of PBP. (Severity: 4)  
Private subroutine/method '\_parse\_first\_line' declared but not used at line 206, column 1.  
Private subroutine/method '\_parse\_body' declared but not used at line 226, column 1.  
Always unpack @\_ first at line 244, column 1. See page 178 of PBP. (Severity: 4)  
Reused variable name in lexical scope: \$value at line 264, column 13. Invent unique names.  
Regular expression without "/x" flag at line 368, column 29. See page 236 of PBP.  
Regular expression without "/x" flag at line 376, column 46. See page 236 of PBP.  
Subroutine "\_build\_cookie" does not end with "return" at line 395, column 1. See page 197 of PBP.

Вопросы?

### 3. Perl-сообщество

# PM-группы – Perl Mongers (Perl-дилеры)



pm.org

[Zurich.pm](#)

**The Republic of Belarus**

[Minsk.pm](#)

**Ukraine**

[Kiev.pm](#)

[Odessa.pm](#)

**United Kingdom**

[Aberdeen.pm](#)

## РМ-встречи



# Конференции



YAPC::Asia YAPC::Brazil YAPC::Europe YAPC::NA  
YAPC::Russia



# YAPC::Europe



YAPC::Europe





# Воркшопы



## Профилірованіє н Релі

Total Elapsed Time = 1299.126 Seconds

Incorporated Time = 437.5927 Seconds

Exclusive Times

nTime	ExclSec	Cpu/s	#Calls	nsCpu/s	Cpu/s
10.5	43.38	42.383	3355440	0.0000	0.0000
8.86	35.75	21.631	187596	0.0002	0.0000
7.15	29.87	33.90	9612	0.0002	0.0000
5.79	24.18	29.351	57243	0.0004	0.0000
4.89	20.43	29.614	450436	0.0009	0.0000
4.10	18.78	29.008	44042	0.0008	0.0000

Без-інтерфейс Релі



Saint Perl  
22 декабря 2012  
Санкт-Петербург

# Хакатоны





Ski Hackathon  
16-20 января 2013  
Карпаты

# Веб-сайты



# The Perl Programming Language

www.perl.org

[HOME](#)
[LEARN](#)
[DOCUMENTATION](#)
[CPAN](#)
[COMMUNITY](#)
[GET INVOLVED](#)
[DOWNLOAD](#)
[ABOUT](#)

## Flexible & Powerful

That's why we love Perl 5

[Get started](#)

[DOWNLOAD PERL](#)



**Current Perl version**  
5.16.2 - [download now](#)

**Find out more**

[Learn](#)  
[Documentation](#)  
[Community](#)  
[Events](#)

**Tip**  
**Object Oriented programming** [Moose](#)®  
(and associated MooseX modules) are a postmodern object system for Perl 5

**Sponsor**  
  
Local Search

 **Learning Perl 5**  
With free online books, over 25,000 extension modules, and a large developer community, there are many ways to learn Perl 5.

 **The Perl Community**  
Perl has an active world wide community with over 300 local groups, mailing lists and support/discussion websites.

 **Contribute to Perl**  
Perl is being *actively developed*. There are many ways to get *involved*.

 **Events and Conferences**  
Conferences, social and technical events around the world offer bits of networking and learning opportunities.

 **The Perl Foundation**®  
The Perl Foundation is dedicated to the advancement of the Perl programming language through open discussion, collaboration, design, and code.

 **Documentation**  
Core documentation, FAQs and translations.

 **CPAN**  
The Comprehensive Perl Archive Net work (CPAN) has over 25,000 open source distributions available for download.

 **Perl 6** ↗  
Perl 6 is a sister language, part of the Perl family. Perl 6 is not production ready yet, you can get involved with its [development](#) though.

**Latest Uploads to CPAN**®

1 - [Carp](#) 4.0.1, [Moose](#) 1.00, [MooseX](#) 0.40.0

2 - [Bless](#) 1.05, [Data](#) 0.70



There's more than one way to blog it.



By **Olivier Mengué (dolmen)** on November 22, 2012 9:23 AM

The port of Git on Win32, [msysgit](#), has Perl bundled with it. This means that most Git users on Windows have a perl installed somewhere. This is an important opportunity to bring developer tools written in Perl to a larger audience. My own [github-keygen](#) tool is one of them.

Unfortunately, that perl has some quirks:

- this is an old 5.8.8, with a huge patch
- it is built on `sys`, which is a quite uncommon environment for Win32 perl developers (from the Perl developer it more like Unix: forward slashes, PATH separator is `;`...)
- some core modules are missing. Noticed in particular the whole `Pod`: I've [tried to work on the Perl upgrade](#), but this is a too tough task for me.

So please help to bring a modern Perl to the Windows world!

1 comment

[Permalink](#)



By **Jeff Thhammer** on November 21, 2012 8:09 PM

[Pinto](#) is a turnkey solution for managing custom repositories of Perl modules. [Stratopan](#) is a hosted service built upon Pinto. Both of them help you take control of your dependency stack by providing a coherent set of tools for managing upgrades and reducing risks.

I'll be giving presentations on Pinto and Stratopan to the Los Angeles Perl Mongers on December 11 at 7:00, and to the Thousand Oaks Perl Mongers on December 12 at 7:00. So if you happen to be in the area, please do stop by – I'd love to see you.



blogs.perl.org is a common blogging platform for the Perl community. Written in Perl and offering the modern features you've come to expect in blog platforms, the site is run by Dave Cross and Aaron Crane, with a design donated by Six Apart, Ltd.





Clear question sand runnable code  
get the best and fastest answer



Search

PerlMonks

## The Monastery Gates

[Log in](#) | [Create a new user](#) | [The Monastery Gates](#) | [Super Search](#)  
[Seekers of Perl Wisdom](#) | [Meditations](#) | [PerlMonks Discussion](#)  
[Obfuscation](#) | [Reviews](#) | [Cool Uses For Perl](#) | [Perl News](#) | [Q&A](#)  
[Tutorials](#)  
[Poetry](#) | [Recent Threads](#) | [Newest Nodes](#) | [Donate](#) | [What's New](#)

(#131) - [suspend](#) ([print](#) / [reply](#) / [xml](#))

Donations gladly accepted

If you're new here please read [PerlMonks FAQ](#)  
and [Create a new user](#).

Want Mega XP? Prepare to have your hopes dashed, join in on the: [perl idiosyncrasy 2012](#) (Don't worry; you've got plenty of time.)

### New Questions

Where should (or could) a distribution override HARNESS\_OPTIONS?

No replies — [Read more](#) | [Post response](#)

by davidg  
on Nov 22, 2012 at  
13:32

I think that increasingly we're seeing people running distribution test suites in parallel by setting the environment variable `HARNESS_OPTIONS=-jN` or by executing `prove -jN`. [chromatic](#) discusses it in his blog [here: Parallelism in Test Suites](#). In that post, and in the discussion that follows, module authors are encouraged to be mindful of parallelism in testing so that in the vast majority of the cases where parallelism is possible module tests are designed to facilitate that capability. An example is given where two test scripts depend on the same clean slate, and where running in parallel they stomp on each other's sandbox. In most such cases, fixing the assumptions or creating independent resources for each test script is a reasonable fix.

In a followup to the same post, [aristotle](#) makes the following statement:

*Making a test suite not break under parallelism doesn't necessitate making it run in parallel. However I believe something like say 95% of tests on CPAN will already run fine in parallel with no further ado, and of the rest, easily the majority will be very simple to fix.*

### Log In

Username:

Password:

☐ remember me

[What's my password?](#)  
[Create A New User](#)

### Chatterbox

and all is quiet...

[How do I...?](#) | [Other CB](#)  
[chats](#)

### Other Users

Others chatting in the  
Monastery: (8)

[Chatterbox](#)

[SuperSearch](#)

[Monks](#)

[Kernels](#)

[gods](#)

[Kustodian](#)

[dmg](#)

[KarlGuthrie](#)

As of 2012-11-22 19:41 GMT

### Sections

[Seekers of Perl Wisdom](#)

[Cool Uses For Perl](#)

[Meditations](#)

[PerlMonks Discussion](#)

# Агрегаторы

# Perlsphere

The Perl firehose! The Web's biggest collection of Perl 5 blogs.  
If you'd like your Perl 5 blog or tech blog's Perl category to appear here, send mail to [leo@cuckoo.org](mailto:leo@cuckoo.org)  
(Please have several posts already).

## Modern Perl Blog: How Bugs Get Fixed

To write a robust program, you must manage the small details. Many of these details you will only discover when real people use your program with real data. A program becomes robust only when it can handle unanticipated cases sensibly and successfully.

If and when you *evaluate a free software project for the first time*, you have the opportunity to help that program become more robust, especially if the platform or data or tasks you have in mind are sufficiently different from those the authors have already explored.

In other words, if you find bugs, the authors may not be aware of them. If you do not report them, they may not get fixed.

I hesitate to suggest that you have an ethical obligation to report bugs, especially if you end up not using the software, but you ought to consider that unreported bugs may never get fixed.

(When I tried to revive the [Perl SDL](#) project several years ago, I couldn't find any one willing or able to try to build the software on Windows and report back debugging information, even though I could find a handful of people willing to say "It doesn't work on Windows". As a consequence, the Perl SDL bindings only started to work on Windows thanks to the superheros of Karik, Takore and the other contributors.)

Note again that I did not write that you have an obligation to report bugs. Nor did I write that you cannot complain about software unless you have reported bugs. I merely wrote that unreported bugs tend to remain unfixed.

Posted by chromatic on free software, perl, software development at Tue, 20 Nov 2012 16:58:12 -0000 | [Permalink](#)

## Perlbuzz: Perlbuzz news roundup for 2012-11-19

These links are collected from the [Perlbuzz Twitter feed](#). If you have suggestions for news bits, please mail me at [andy@perlbuzz.com](mailto:andy@perlbuzz.com).

- Help fire up this year's Google Code-In ([news.perlboundation.org](#))
- How to work with comma-separated and tab-separated data ([perltraining.com.au](#))
- Module::Build::Clean::Install deletes old versions first ([blogs.perl.org](#))
- p5p summary for last week ([blogs.perl.org](#))
- Help name a new module for IPv6 subnets ([blog.uth.org](#))
- Serial is a new binary data serialization format ([biop.binding.org](#))
- Promoting Perl workshops and conferences ([blogs.perl.org](#))
- Filtering output with Template Toolkit ([blogs.perl.org](#))
- Help with a Perl project layout problem ([jascroll.com](#))

## Subscriptions

[Perl News](#)  
[Perl.com](#)  
[Perl5 Porters](#)  
[Summaries](#)  
[Aaron Crane.co.uk](#)  
[Aristotle](#)  
[The Cattle Grid](#)  
[cherny](#)  
[Code and Hacks](#)  
[dagolden](#)  
[Gabor Szabo](#)  
[Damen Learn](#)  
[Programming](#)  
[Dave's Free Press](#)  
[Journal](#)  
[dukeletapl](#)  
[House Absolute\(Ly](#)  
[Pointless\)](#)  
[Jérôme Quelin](#)  
[logicLAB.org](#)  
[MathsGSL / PerlGnu](#)  
[Scientific Library](#)  
[Lyle's Perl Blog](#)  
[Mark Stenberg](#)  
[Miyagawa's Journal](#)  
[Modern Perl Blog](#)  
[Rangward's Blog](#)  
[Peter Sergeant](#)  
[Eric Johnson \(hablamo\)](#)  
[Moving to Moose](#)  
[hackerthon](#)  
[More Room In The Room](#)  
[nothingmuch's perl blog](#)  
[rnx's Perl](#)  
[The Onion Stand](#)  
[Ovid](#)  
[patspam.com](#)  
[PerlBuz](#)  
[The Perl Hacker Painter](#)  
[scriptivity.org](#)  
[Perlspeak.de](#)  
[Perl Ideas](#)  
[Phread](#)  
[Ricardo Signes](#)  
[sarlak](#)  
[Schwern](#)  
[Shlomi Fish](#)

# \$perl->buzz

What's happening in the world of Perl programming, including Perl5, Perl6, the CPAN and Parrot!

## Perlbuzz news roundup for 2012-11-19

By Andy Lester on November 19, 2012 10:32 AM | [10 Comments](#)

These links are collected from the [Perlbuzz Twitter feed](#). If you have suggestions for new b.b.s, please mail me at [andy@perlbuzz.com](mailto:andy@perlbuzz.com).

- How to work with comma-separated and tab-separated data ([perlfaq.org](#))
- Module: Build C and install deleted or diverged as first ([blogs.perl.org](#))
- p5p summary for last week ([blogs.perl.org](#))
- Help name a new module for Thy subverts ([blogs.perl.org](#))
- Serialise a new binary data serialisation format ([blogs.perl.org](#))
- Promoting Perl workshops and conferences ([blogs.perl.org](#))
- Filtering output with Template Toolkit ([blogs.perl.org](#))
- Help with a Perl project layout problem ([blogs.perl.org](#))
- Secure has LWP backend ([blogs.perl.org](#))
- Hello distributed job processing system for Perl updated ([blogs.perl.org](#))
- Why PDB, UNICORE makes me SAD ([blogs.perl.org](#))
- Help @\_fix with this year's Perl Advent Calendar ([perladvant.org](#))
- Using Padre for the first time ([blogs.perl.org](#))

## Follow Perlbuzz

twitter

## Job hunting for programmers



Land the Tech Job You Love, Andy Lester's guide to job hunting for programmers and other technical professionals, is available in PDF, ePub and audiobook formats, all \$29.95, as well as good old-fashioned paper.

## Perlbuzz

Perl buzz news roundup for 2012-11-19  
Perl buzz news roundup for 2012-11-12  
Perl buzz news roundup for 2012-11-05  
Perl buzz news roundup for 2012-10-29  
Perl buzz news roundup for 2012-10-22  
Perl buzz news roundup for 2012-10-15  
Perl buzz news roundup for 2012-10-08  
Perl buzz news roundup for 2012-10-01  
Perl buzz news roundup for 2012-09-24  
Perl buzz news roundup for 2012-09-17  
Perl buzz news roundup for 2012-09-10  
Perl buzz news roundup for 2012-09-03  
Perl buzz news roundup for 2012-08-27

## Contact

Email [andy@perlbuzz.com](mailto:andy@perlbuzz.com),  
Twitter [EzraLester](#)

## Other Perl Sites

[perl.org](#)  
[rakudo.org](#), home of Rakudo Perl 6  
[Moose Perl](#)  
[Perl To The Perl](#)  
[The Official Perl Wiki](#)  
[Perl 5](#)  
[Perl 6](#)  
[Perl6](#)  
[News at The Perl Foundation](#)  
[www.perl.org](#)  
[perlfaq.org](#)  
[Perl.com](#), the Perl website  
[Yat, Another Perl Gimmick](#)  
[Blackhat](#)

## Archives

November 2012  
October 2012  
September 2012  
August 2012  
July 2012  
June 2012  
May 2012  
April 2012  
March 2012  
February 2012  
January 2012  
December 2011  
November 2011  
October 2011  
September 2011  
August 2011  
July 2011  
June 2011  
May 2011  
April 2011

## Perlbuzz news roundup for 2012-11-12

By Andy Lester on November 12, 2012 9:03 AM | [10 Comments](#)

These links are collected from the [Perlbuzz Twitter feed](#). If you have suggestions for new b.b.s, please mail me at [andy@perlbuzz.com](mailto:andy@perlbuzz.com).

- Help fix a this year's Google Code-In ([blogs.perl.org](#))
- How to work with comma-separated and tab-separated data ([perlfaq.org](#))
- Module: Build C and install deleted or diverged as first ([blogs.perl.org](#))
- Sparktime in Excel: Writer & XLSX ([blogs.perl.org](#))

## Perlbuzz news roundup for 2012-11-05

By Andy Lester on November 5, 2012 9:36 AM | [10 Comments](#)

These links are collected from the [Perlbuzz Twitter feed](#). If you have suggestions for new b.b.s, please mail me at [andy@perlbuzz.com](mailto:andy@perlbuzz.com).

- A view of representation in the Perl garbage collection ([blogs.perl.org](#))
- Easy XML parsing with XML::LibXML ([blogs.perl.org](#))
- A new release of Caracal: sort: Move for your ascending to insertion ([blogs.perl.org](#))
- Perl 5.16.2 released ([blogs.perl.org](#))
- Venerating Perl 5's release history ([blogs.perl.org](#))
- Strawberry Perl 5.16.2.1 released ([blogs.perl.org](#))



# Perl Weekly

A free, once a week e-mail round-up of hand-picked news and articles about Perl.

*You are busy churning out code or managing the developers. You care about Perl but don't have time to go through tens and hundreds of articles and blog posts every day. You want to keep an eye on the development of Perl without drowning in a sea of blog posts. You need someone to point out the most important news and articles in the Perl World.*

*Let me be your guide.*

Join the 4057 people who received the previous issue!

Enter your e-mail:

**and click to sign up!**

Just ONE e-mail each Monday. Easy to unsubscribe. No spam. Your e-mail address is safe.

Next issue on November 26, 2012

Want a sneak peek? Check out the [latest](#) issue or see [every earlier issue](#).

Hand-picked by [Gabor Szabo](#)  
long time Perl developer and  
trainer. White camel awardee.



## What the Readers Say



"The Perl Weekly is an excellent aggregation of all Perl things bbb and news-worthy. I follow a lot of blog posts, but I find this weekly recap helps me find interesting stuff I may have missed."

**Olaf Alders (oalders)**  
Perl developer, Owner of [Wunder Counter](#), Creator of [MetaCPAN](#).



"Thanks for putting together the newsletter. It's a helpful resource."

**Richard Dice**  
[@richarddice](#), president of [The Perl Foundation](#), from Aug 2007 till 2009.

# Русский Перл

Что пишут о Перле на русском языке

## М Е Т А

Последний обновлен было  
2012-11-22 в 21:42 по Москве  
[Добавить сайт](#)

## П О И С

## У Ч А С

[Сообщество ru.perl в LiveJournal](#)

[Сообщество ru.catalyst в LiveJournal](#)

[Moscowperl 2.0](#)  
[perl5.ru](#)

[Антоний Шарифуллин](#)

[On Perl — блог о языке программирования Perl и его сообществах](#)

[Мини-портал Perl на Opennet](#)

[bimagaibov](#)

[laziness, impatience and Hubris](#)

[Язык Perl | Сообщества на U.VISION.RU](#)

[Internal Server Error](#)

[XPerl.ru |](#)

[Программирование : Perl :: Ooo](#)

[XPerl.ru |](#)

[Программирование : Perl :: Moz](#)

[XPerl.ru |](#)

[Программирование : Perl :: Web-а-сценки](#)

[XPerl.ru |](#)

[Программирование : Perl :: Perl-выражения](#)

[XPerl.ru |](#)

[Программирование : Perl :: Perl-это](#)

## Четверг, 22-е ноября

### [allframeworks / Perl: MurmurHash - новая хэш-функция в Perl 5.17.6](#)

Существенным изменением в Perl 5.17.6 стала установка по умолчанию новой хэш-функции MurmurHash (v3) и изменение по-настоящему лучшего случайного порядка следования ключей в хэш-таблице (ас-социативном массиве). Изменение прошло как-то обыденно и незаметно, но думаю, что это будет достаточно заметной фишкой будущего стабильного релиза.

Написал хота в 09:40

## Среда, 21-е ноября

### [Сообщество ru.perl в LiveJournal: Вакансии в Ростове-на-Дону](#)

Приветствую, сообщество.

Памятуя об том, как здешние обитатели однажды помогли найти одному из индустриалу замечательного спеца, обращаюсь к вам снова.

Для крупного конструктора сайтов нужны **perl-разработчики** в офисе в Ростове-на-Дону.

**И надо** - Perl изи PHP/C++ с готовностью переучиваться на Perl (расматриваем без опыта, будем обучать); шарить в mysql, css, js, jquery, ajax.

Плюшки, интересности и прочее прилагаются. Детали готовы обсуждать.

Если кто-то из **вы** таких знаете, пишите: [nikbotova@ichar.ru](mailto:nikbotova@ichar.ru), skypes: lex.lwell

Всем спасибо.

Написал Alex ndra в 16:15

## Вторник, 20-е ноября

### [Perlito: Глава 7. Горячие веб-топики](#)

22:30 20.11.2012

#### [Глава 7. Горячие веб-топики](#)

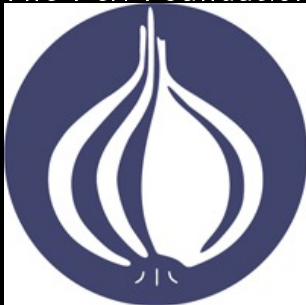
Традиционно веб-приложения одиозны и изолированы. Каждая из них имеет собственный набор пользователей и данных, и никогда не признают существования других приложений. Сегодня ситуация меняется. Веб-приложения объединяются, обмениваются данными.

IRC

irc.perl.org – #ru.pm!

# Организации

# The Perl Foundation



# Enlightened Perl Organization



# YAPC Europe Foundation

