

**Знакомство**

**с Perl 6**



220 слайдов

Про Perl 6 говорят с

2000

perl6.ru появился в

2003

Это мой 7-й рассказ с

2007

Сейчас

2009

ГОТОВ ли Perl 6?

ГОТОВ ли Perl 6?

Ну почти

Когда бюджет?

Когда будет?

На Рождество

Когда будет?

На Рождество

(год не указан)

**pre-alpha**

**2009-2010**

Когда в production?

Когда в production?

**ЗАВИСИТ ОТ НАС**

Когда в production?

**ЗАВИСИТ ОТ ВАС**

Когда в production?

Уже сейчас!

Когда в production?

Уже сейчас!

(На самом деле года через два после релиза)

Зачем вообще нужен

Perl 6?

**КТО ПОМНИТ ПЕРЕХОД**

**с Perl 4 на Perl 5?**

$$4! = 5$$

$$5 \neq 6$$

5! = 6

Perl = Perl

`'Perl' eq 'Perl'`

Как изучать

Perl 6?

**Где Взять**

**Perl 6?**

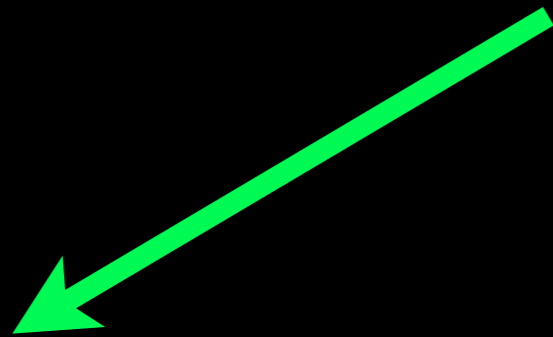
**Можно ли попробовать**

**Perl 6?**

?

**Как изучать**

# Как изучать



Знать Perl 5

изучить  
отличия

# Как изучать

```
graph TD; A[Как изучать] --> B[Знать Perl 5]; A --> C[Не знать Perl 5];
```

Знать Perl 5

Не знать Perl 5

изучить  
отличия

# Как изучать

```
graph TD; A[Как изучать] --> B[Знать Perl 5]; A --> C[Не знать Perl 5];
```

**Знать Perl 5**

**изучить  
отличия**

**Не знать Perl 5**

**изучить  
документацию**

**Изучать ли**

**ламповую схемотехнику,  
когда есть транзисторы?**

**У нас есть два языка!**

**У нас есть два языка!**

**Perl 5 и Perl 6**

**У нас есть два языка!**

**Perl 5 и Perl 6**

**Завоевание мира стало проще**

# Отличия от Perl 5

# Отличия от Perl 5 (не все)

# Отличия от Perl 5

# Отличия от Perl 5

**В идеологии**

**Синтаксис Perl 6  
описан грамматикой**

**Синтаксис Perl 6  
описан грамматикой**

**Синтаксис Perl 5  
задан реализацией**

**Много реализаций**

**(пока нет ни одной завершенной)**

**Одна реализация**

**(портирована на многие платформы)**

Perl 6 компилируется

**Perl 6 компилируется**

**Perl 5 интерпретируется**

Программа

Компилятор

Байт-код

Виртуальная машина



Программа

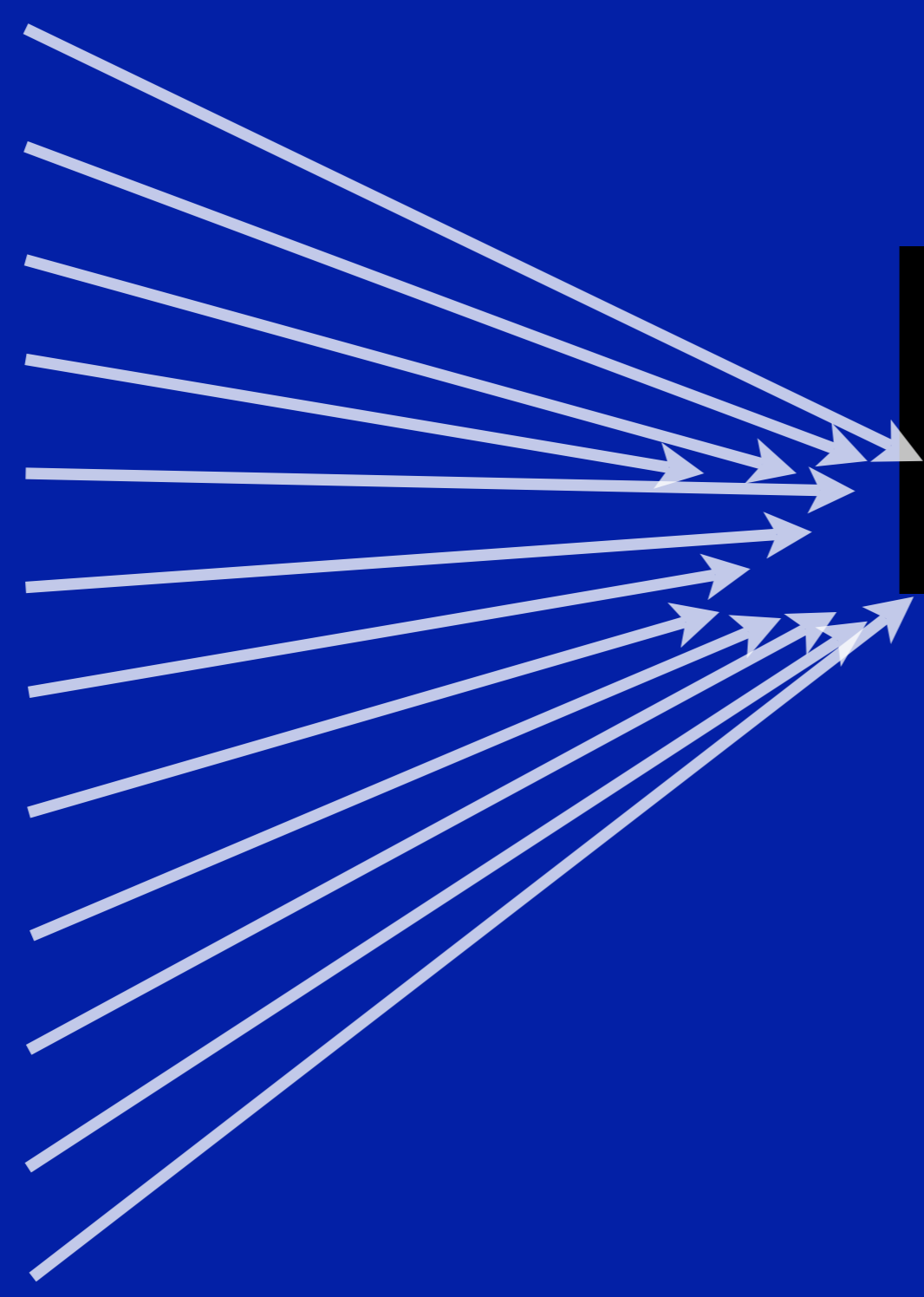
Компилятор

Байт-код

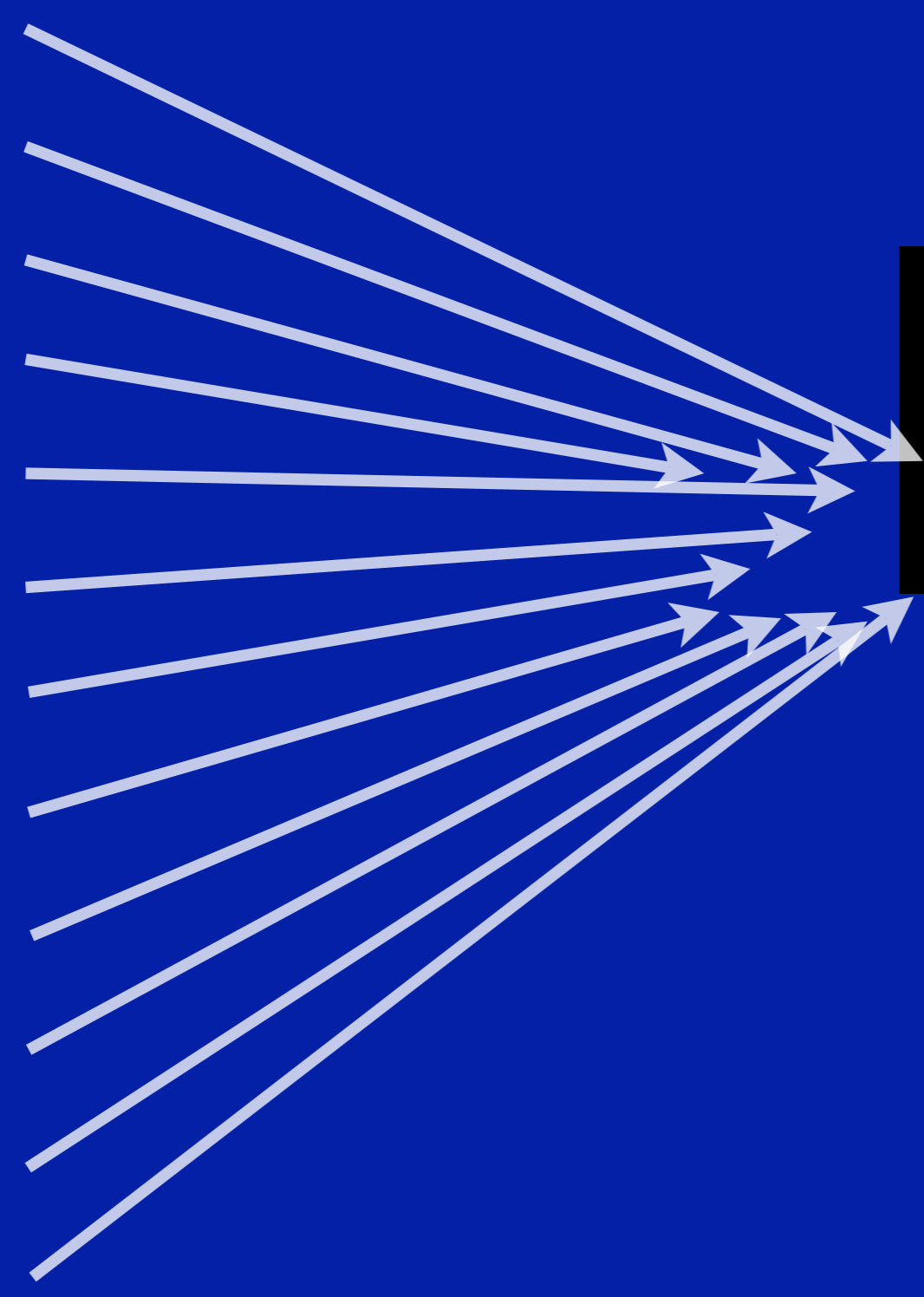


Виртуальная машина

Basic  
Forth  
Jako  
Lisp  
m4  
Ook  
Perl 6  
Perl 5  
Python  
Ruby  
Scheme  
Tcl

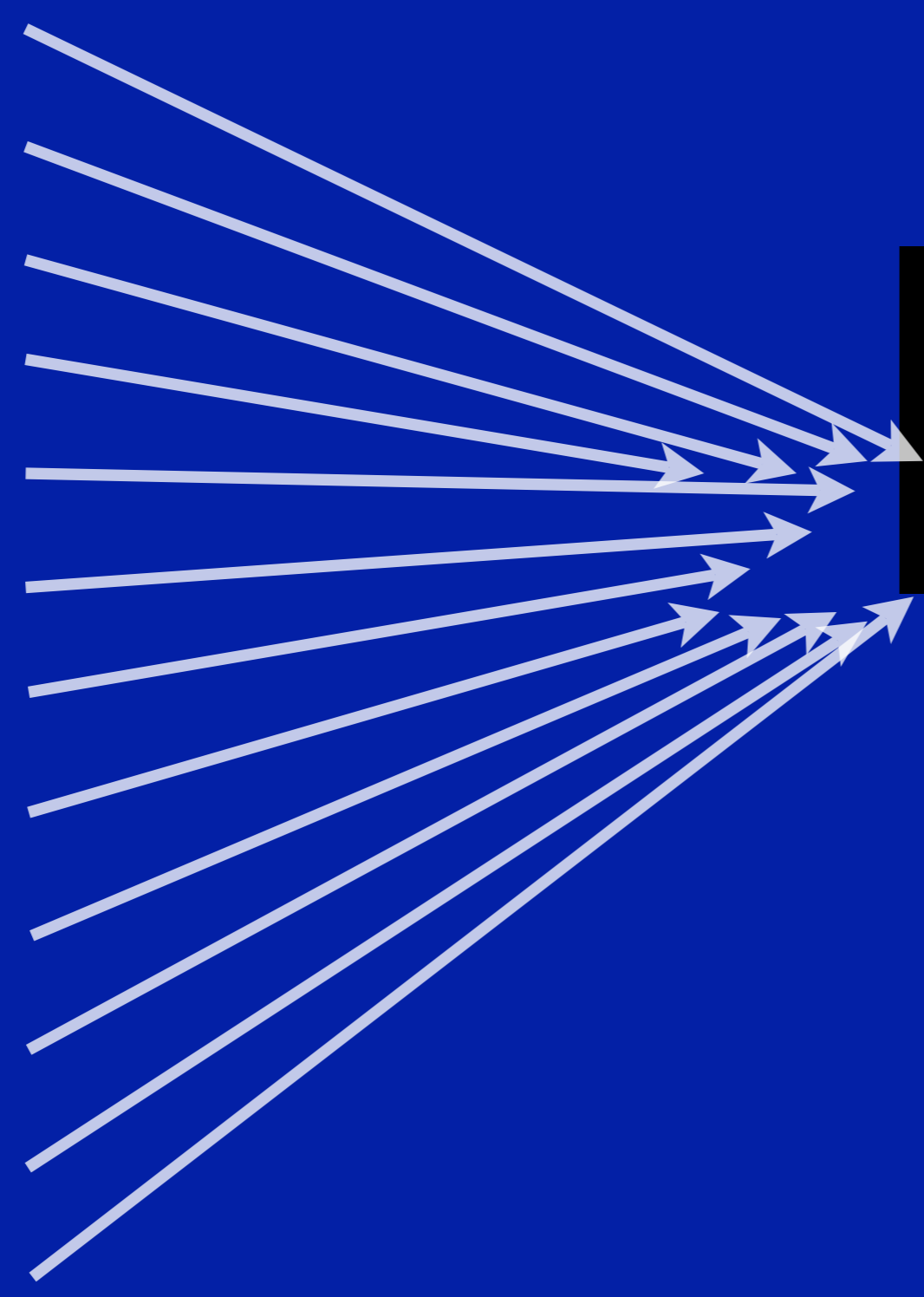


Basic  
Forth  
Jako  
Lisp  
m4  
Ook  
Perl 6  
Perl 5  
Python  
Ruby  
Scheme  
Tcl



байт-код

Basic  
Forth  
Jako  
Lisp  
m4  
Ook  
Perl 6  
Perl 5  
Python  
Ruby  
Scheme  
Tcl



LOL-КОД

# Отличия от Perl 5

**В синтаксисе**

# *Хафрфманизизация*

# Perl 5

```
print "Hello, World!";
```

# Perl 6

```
say "は、ラクダの方法";
```

# Perl 6

```
"は、ラクダの方法".say;
```

# Perl 6

```
my $ναρταβλε =  
    "は、ラクダの方法";
```

# Perl 6

```
$str.chars;
```

```
$str.bytes;
```

```
$str.glyphs;
```

# Perl 6

```
$str.chars.say;
```

# Perl 6

```
$str.chars.say;
```

```
3.14.elems.say;
```

*Унификация*

**СИГИЛЫ**

# Perl 5

```
my @arr = (0..5);
```

```
say $arr[2];
```

# Perl 6

```
my @arr = (0..5);
```

```
say @arr[2];
```

# Perl 6

```
my @arr = (0..5);
```

```
say @arr[2, 3];
```

# Perl 6

```
my @arr = (0..5);
```

```
say @arr.elems;
```

# Perl 5

```
my %hash = (  
    city => "Tashkent",  
    year => 2009  
);  
say $hash{city};
```

# Perl 6

```
my %hash = (  
    city => "Tashkent",  
    year => 2009  
);  
say %hash<city>;
```

# Perl 6

```
my %hash =  
    city => "Tashkent",  
    year => 2009  
;  
say %hash<city year>;
```

# Perl 6

```
my %hash =  
    city => "Tashkent",  
    year => 2009  
;  
say %hash.kv;
```

**Сигилы — не тип переменной**

**Сигилы — интерфейс контейнера**

@

Объект реализует роль `Positional`  
и поддерживает метод  
`postcircumfix:<[ ]>`

%

Объект реализует роль *Associative*  
и поддерживает метод  
`postcircumfix:<{ }>`

# &

Объект реализует роль Callable  
и поддерживает метод  
postcircumfix:<( )>

qw

# Perl 5

```
my @cities = (  
    "Moscow", "Kiev", "Tashkent"  
);
```

# Perl 6

```
my @cities =  
    "Moscow", "Kiev", "Tashkent"  
;
```

# Perl 5

```
my @cities = qw(Moscow Kiev Tashkent);
```

# Perl 6

```
my @cities = <Moscow Kiev Tashkent>;
```

**ТВИГИЛЫ**

# Perl 5

```
$ENV{SERVER_NAME}
```

# Perl 6

```
%*ENV<SERVER_NAME>
```

# Perl 6

`%*ENV<SERVER_NAME>`

# Perl 6

```
%*ENV<SERVER_NAME>
```

# Perl 6

```
%*ENV<SERVER_NAME>
```

`$var`

`$.var`

`^var`

`:var`

`*var`

`?var`

`=var`

`!var`

# Perl 5

\_\_LINE\_\_

\_\_FILE\_\_

\_\_PACKAGE\_\_

# Perl 6

`$?LINE`

`$?FILE`

`$?PACKAGE`

# Цепочки сравнений

# Perl 5

```
if ($latitude > 41.2 &&  
    $latitude < 41.4) {  
    print "Tashkent\n";  
}
```

```
if 41.2 < $latitude < 41.4 {  
    say "Tashkent";  
}
```

**Контексты**

# Строковый контекст

```
my @array = (5..10);
```

```
say ~@array;
```

```
5 6 7 8 9 10
```

# Числовой контекст

```
my @array = (5..10);
```

```
say +@array;
```

6

# Строковый контекст в контексте хеша

```
my @array = (5..10);
```

```
say ~ hash @array;
```

5 6

7 8

9 10

# Булевый контекст

```
my @array = (5..10);
```

```
say ?+@array;
```

# Конкатенация строк

```
print
```

```
    "Uzbek "
```

```
    . "Perl "
```

```
    . "Workshop";
```

```
print
```

```
    "Uzbek "
```

```
~  "Perl "
```

```
~  "Workshop";
```

# Установка Rakudo

# Исключения

# Perl 5

```
eval {  
    print 1/0;  
}  
if ($?) {  
    print_error();  
}
```

# Perl 6

```
try {  
    say 1/0;  
  
    CATCH {  
        print_error();  
    }  
}
```

# Perl 6

```
try {  
    say 1/0;  
  
    CATCH {  
        print_error();  
    }  
}
```

# ЦИКЛЫ

# Perl 5

```
foreach (@list) {  
    . . .  
}
```

# Perl 5

```
for (@list) {  
    . . .  
}
```

# Perl 6

```
for @list {  
    . . .  
}
```

# Perl 6

```
for @list -> $x {  
    say $x;  
}
```

# Perl 6

```
for @list -> $x {  
    say $x;  
}
```

# Perl 6

```
for @list -> $x, $y {  
    say $x + $y  
}
```

# Perl 6

```
loop(my $t = 300; $t; $t--){  
    sleep 1;  
};
```

# ФУНКЦИИ

```
sub flight($from, $to) {  
    say "$from --> $to";  
}
```

```
flight("Москва",  
      "Ташкент");
```

```
sub flight($from, $to) {  
    say "$from --> $to";  
}
```

```
flight("Москва",  
      "Ташкент");
```

```
sub flight($from, $to) {  
    say "$from --> $to";  
}
```

```
flight "Москва",  
      "Ташкент";
```

```
sub flight($from, $to) {  
  say "$from --> $to";  
}
```

```
flight  
  from => "Москва",  
  to   => "Ташкент";
```

```
sub flight($from, $to) {  
  say "$from --> $to";  
}
```

```
flight  
  from => "Москва",  
  to   => "Ташкент";
```

```
sub choose(@a, @b) {  
    return  
        any(@a) < all(@b) ??  
        1 !! 2;  
}
```

```
say choose(  
    @SkyExpressPrices,  
    @AeroflotPrices  
);
```

# Гипероператоры

```
my @Flight = (...);
```

```
my @Food = (...);
```

```
my @Flight = (...);
```

```
my @Food = (...);
```

```
my @FullPrice =  
  @Flight >>+<< @Food;
```

```
my @Flight = (...);
```

```
my @Food = (...);
```

```
my @FullPrice =  
  @Flight >>+<< @Food;
```

```
@FullPrice >>+=<< $Fuel;
```

```
my @Flight = (...);
```

```
my @Food = (...);
```

```
my @FullPrice =  
  @Flight »+« @Food;
```

**Объединения**

**Объединения**

(квантовые  
суперпозиции)

```
sub choose(@a, @b) {  
  return  
    any(@a) < all(@b) ??  
    1 !! 2;  
}
```

```
say 1  
  if 20 == 10 | 20 | 30;
```

```
say 1  
  if 20 == 10 | 20 | 30;
```

```
say 1  
  if 20 == any(10, 20, 30);
```

```
say 1  
  if 20 == none(1, 2, 3);
```

```
my $j = 2009 | 2010;
```

```
say 1 if 2009 == $j;
```

# Перегрузка операторов и функций

# Перегрузка операторов и функций

(множественная диспетчеризация)

$$1 + 2$$

$$2.71 + 3.14$$

```
multi sub
```

```
    elections($candidate);
```

```
multi sub
```

```
    elections(@candidates);
```

```
multi infix:<+> ($a, $b) {  
    return $a - $b;  
}
```

```
say 10 + 20;           -10
```

```
multi postfix:<!> ($n) {  
    return [*] 1..$n;  
}
```

```
say 2009!;
```

```
multi postfix:<!> ($n) {  
    return [*] 1..$n;  
}
```

```
say 2009!;
```

**Где ВЗЯТЬ**

# Предыстория

# Parrot 0.1.0

languages/per16

# Parrot 0.1.0

Для компиляции файла  
грамматики Perl 6  
нужен 1 ГБ памяти

# Parrot 0.1.0

Настоящий байткод

[real.per16.ru](http://real.per16.ru)

# PUGS

PUGS

Perl User's Golfing System

PUGS

Perl User's Golfing System

Написан на Haskell

# PUGS

## Perl User's Golfing System

Вместе с GHC собирается  
часа четыре

**PUGS**

**Perl User's Golfing System**

**Самое большое  
ПОДМНОЖЕСТВО ЯЗЫКА  
НА ТОТ МОМЕНТ**

```
use v6;
```

```
use v6;
```

**Модули на CPAN для Perl 5**

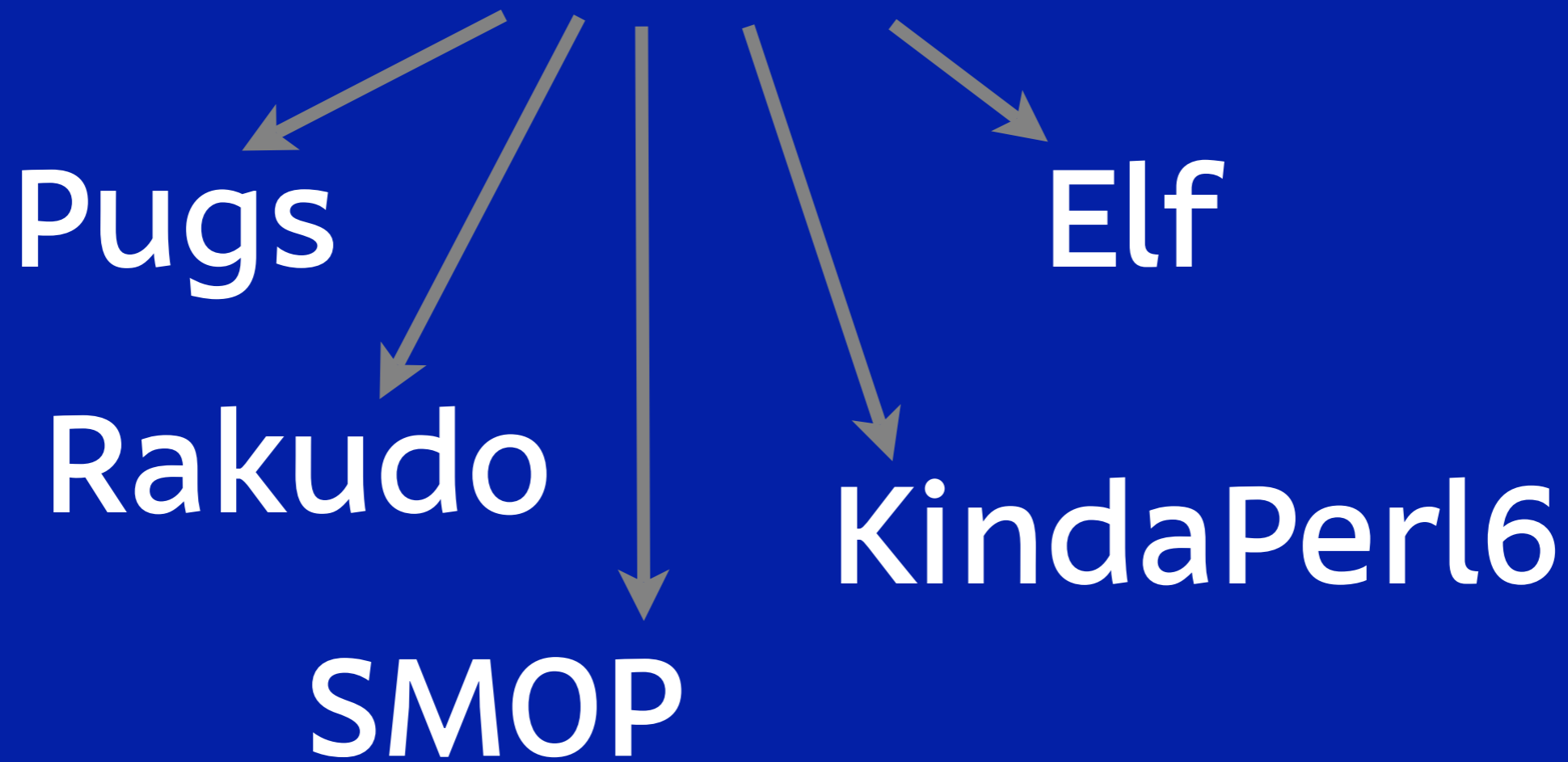
use v6;

Moose?

use v6;

5.10?

# Perl 6 не один



# Rakudo

# Rakudo

Rakudo-do —  
путь верблюда (*яп.*)

# Rakudo

## Компилятор в PIR-код

# Установка Rakudo

```
$ git clone git://github.com/rakudo/rakudo.git
```

# Установка Rakudo

```
$ git clone git://github.com/rakudo/rakudo.git
```

```
$ cd rakudo
```

# Установка Rakudo

```
$ git clone git://github.com/rakudo/rakudo.git
```

```
$ cd rakudo
```

```
$ perl Configure.pl --gen-parrot
```

# Установка Rakudo

```
$ git clone git://github.com/rakudo/rakudo.git
```

```
$ cd rakudo
```

```
$ perl Configure.pl --gen-parrot
```

# Установка Rakudo

```
$ git clone git://github.com/rakudo/rakudo.git
```

```
$ cd rakudo
```

```
$ perl Configure.pl --gen-parrot
```

```
$ make
```

# Установка Rakudo

Сегодня нет `make install`

# Установка Rakudo

```
$ ./per16 -v
```

# Установка Rakudo

```
$ ./perl6 -v
```

```
This is Rakudo Perl 6, revision 38679 built on  
parrot 1.1.0-devel  
for darwin-thread-multi-2level.
```

```
Copyright 2006-2008, The Perl Foundation.
```

# Компилятор Rakudo

[rakudo.org](http://rakudo.org)

# Виртуальная машина Parrot

[parrot.org](http://parrot.org)

# Документация, тесты и грамматика

[pugscodex.org](http://pugscodex.org)

# Веб-приложения на Perl 6

real.perl6.ru

2004 год

```
int main() {  
    char* argv[] = {  
        "parrot",  
        getenv ("PATH_TRANSLATED"),  
        NULL  
    };  
    execvp("/parrot-0.1.0/parrot", argv);  
    return 0;  
}
```

```
my @keys = (  
    'SERVER_NAME',  
    'REMOTE_ADDR',  
    'HTTP_USER_AGENT'  
);  
my $key;  
foreach $key (@keys){  
    print1 "$key=%ENV{$key}<br />";  
}
```

```
my @keys = (  
    'SERVER_NAME',  
    'REMOTE_ADDR',  
    'HTTP_USER_AGENT'  
);  
my $key;  
foreach $key (@keys){  
    print1 "$key=%ENV{$key}<br />";  
}
```

```
sub params2hash(  
  %params,  
  @params_key,  
  @params_value){  
  for 0 .. @params_key -> $c {  
    %params{@params_key[$c]} =  
      @params_value[$c];  
  }  
}
```

```
.include "counter.imc"
```

```
open      P0, "counter.txt"
```

```
readline S0, P0
```

```
set       I0, S0
```

```
inc       I0
```

```
set       I2, 0
```

```
seek      P0, I2, I2
```

```
print     P0, I0
```

```
close     P0
```

[november-wiki.org](http://november-wiki.org)

2008 год

Рождество — December

Вики на Perl 6 — November

Про что  
не рассказал?

Про классы

Про классы,  
про роли

**Про классы,  
про роли  
и про грамматики**

Кратко об этом  
на примере November

```
#!/perl6
use v6;

use CGI;
use Wiki;

my Wiki $wiki = Wiki.new;
$wiki.init();
my $cgi = CGI.new;
$cgi.init();
$wiki.handle_request($cgi);
```

# Классы и роли

```
class Wiki does Session {  
  my $.template_path      is rw;  
  my $.userfile_path      is rw;  
  
  has Storage $.storage   is rw;  
  has CGI      $.cgi       is rw;  
  
  . . .  
}
```

```
method handle_request(CGI $cgi) {
  $.cgi = $cgi;
  my $action = $cgi.param<action> // 'view';
  given $action {
    when 'view' {
      self.view_page(); return;
    }
    when 'edit' {
      self.edit_page(); return;
    }
    when 'log_in' {
      self.log_in(); return;
    }
  }
  self.not_found();
}
```

```
has Storage $.storage is rw;
```

```
class Storage {
```

```
  . . .
```

```
}
```

```
class Storage::File is Storage {
```

```
  . . .
```

```
}
```

```
$.storage = Storage::File.new();
```

```
$.storage.init();
```

```
my $input = $*IN.slurp();  
self.parse_params(%params, $input);
```

# Грамматика

```
if $par ~~ Text::Markup::Wiki::Minimal::Syntax::parag
{
    . . .
}
else {
    $result = '<p>Could not parse paragraph.</p>';
}
```

```
grammar Text::Markup::Wiki::Minimal::Syntax {
    token paragraph { ^ [<heading> || <parchunk>+] $ };
    token heading { '==' <parchunk>+ '==' };
    token parchunk { <twext> || <wikimark> || <metachar> ||
        <malformed> };
    token twext { [ <.alnum> || <.otherchar> || <.whitespace> ]
        + };
    token otherchar { <[ !..% (../ : ; ? @ \\ ^..` {..~ ]> };
    token whitespace { ' ' | \n };
    token wikimark { '[[<twext>]]' };
    token metachar { '<' || '>' || '&' || '\' };
    token malformed { '[' || ']' }
}
```

grammar

Text::Markup::Wiki::Minimal::Syntax {

}

```
token paragraph {  
    ^ [<heading> || <parchunk>+] $  
};
```

```
token heading {  
    '===' <parchunk>+ '==='  
  
};
```

**STD.pm — 150 КБ**

**Что почитать**

*The Community's Rewrite of Perl*

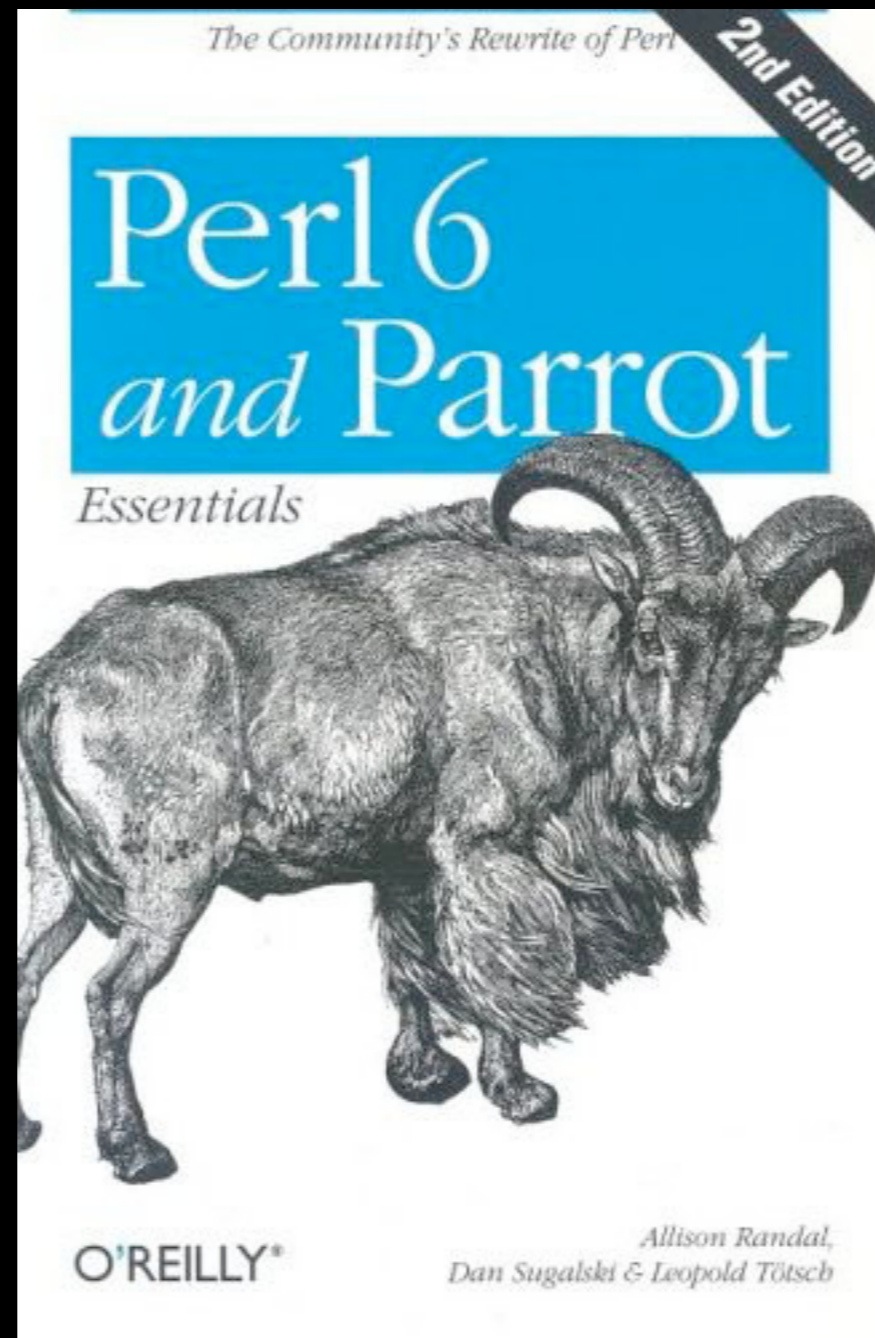
# Perl 6 Essentials



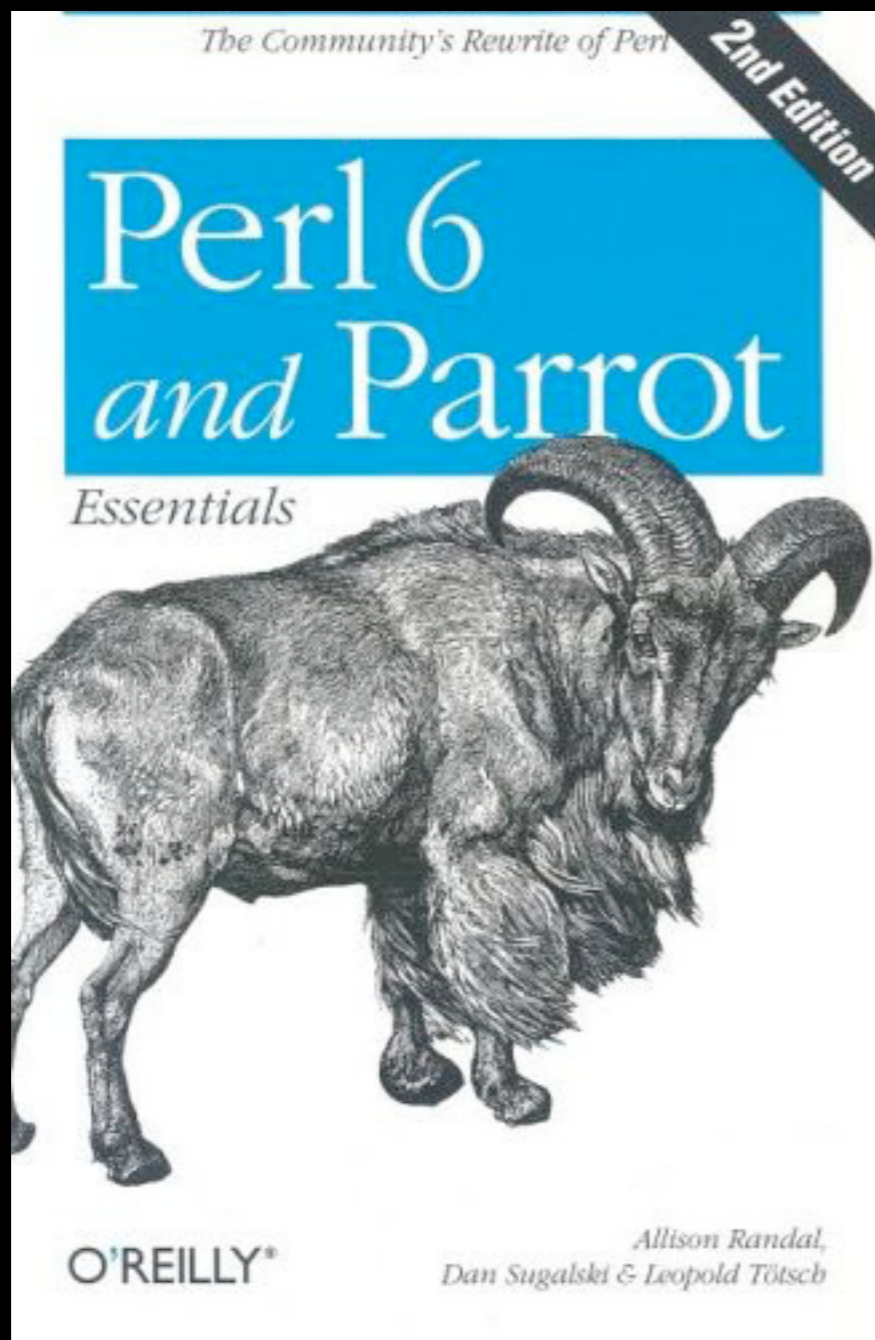
O'REILLY

*Allison Randal & Dan Sugalski*

*Июнь 2003*



Июнь 2004



Июнь 2004



2005

THE EXPERT'S VOICE® IN OPEN SOURCE

# Perl 6 Now

The Core Ideas Illustrated with Perl 5

*Lead the way by mastering Perl 6 techniques now.*

Scott Walters

Copyrighted Material

ADress®

2005

THE EXPERT'S VOICE®



# Beginning Perl 6

From Novice to Professional

James Lee

*2007?*

## Чего ждать от Perl 6



Андрей Шитов

Perl готовится сделать новый рывок и оставить конкурентов позади. До круглой цифры счетчику версий осталось совсем немного, всего 0.1.2 версии.

Про Perl 6 впервые начали говорить в 2000 году, когда было принято решение переписать существующий язык с чистого листа. Исторически сложилось так, что формального стандарта никогда не существовало. Стандартом пятой версии фактически был язык, воплощенный в интерпретаторе. Желание иметь четкие правила — одна из предпосылок к созданию нового языка.

Кроме того, требовались изменения, чтобы, с одной стороны, очистить

язык от устаревших конструкций (например, GLOB), а с другой — добавить «современности» (например, включив в язык полноценную поддержку классов).

### Часть 1. До языка

К сожалению для разработчиков и к счастью для стандарта, после шести лет еще не существует ни окончательных правил, ни полноценного компилятора. Тем не менее, уже сегодня язык можно применять на практике (хо-

тя так будут поступать разве что истинные энтузиасты), а уж познакомиться с языком просто необходимо.

Общее направление задал создатель Perl Ларри Уолл: в статье под общим названием «Откровения» (Apocalypses) описано, каким он видит будущий язык.

Позже появился еще один цикл «Толкования» (Exegeses), в которых Дамиан Конвей поясняет «Откровения» на примерах. И, наконец, существует начатый Эллисон Рэндал набор

«объектов» (Sy...  
...случае спец...  
...подключаютс...  
......  
...из разраб...  
...нового ком...  
...Tang) из...  
...трафике эво...  
...временам...  
...не менее, над...  
...Perl 6 ест...

### Perl 6 и Perl 5

......пятой...  
......друг...  
......начать изу...  
......предва...  
...5. Однако п...  
......пятой ве...  
......многие хо...  
......версию...  
......из преды...  
......только косм...  
...Perl 5 и Perl...  
......одного а...  
......казать, что...  
......обединени...  
......компил...  
......прогр...  
......не менее из...  
......что рабоч...  
......станет...  
......кстати, пре...  
......с четверт...  
......требовал от...  
......усил...  
......известности...

### Архитектура

......Perl...  
......язык...  
......прот...  
......возможности...  
......в...  
......мод...  
......не сл...  
......напри...  
......под мо...  
......специ...  
......ред...  
......работ...  
......более сложно...  
...Perl 6 с...  
......измени...  
......применить с...  
......баз...  
......Эта сх...  
......уже работа...  
......тоже в техн...

Это все уже история

# СЕТЕВЫЕ РЕШЕНИЯ

СПЕЦВЫПУСК

№ 9

сентябрь  
2008 год



## BY Perl

Беларускі Perl-воркшоп 2008

18 октября

<http://event.peribelarus.org/minsk2008/>

**Документация**

**RFCs,**

**Apocalypses,**

**Exegeses**

Все это уже история

RFCs,

Apocalypses,

Exegeses,

**Synopses**

# Synopses

[perlcabal.org/syn](http://perlcabal.org/syn)

# Synopses

[perlcabal.org/syn](http://perlcabal.org/syn)

S1 . . . S33

**S = Synopses**

**S = Spec**

**Кого почитать**

Мориц Ленц

[perlgeek.de](http://perlgeek.de)

Габор Сабо

[szabgab.com](http://szabgab.com)

Джонатан  
Ворthingтон

[use.perl.org/](http://use.perl.org/)

~JonathanWorthington

**Карл Мэсак**

`use.perl.org/~masak`

Патрик Мишо

[use.perl.org/~pmichaud](http://use.perl.org/~pmichaud)

**Илья Беликин**

[ihrd.livejournal.com](http://ihrd.livejournal.com)

perl6.ru

**\_\_\_END\_\_\_**

**Андрей Шитов**

**<http://shitov.ru> | [andy@shitov.ru](mailto:andy@shitov.ru)**