

Solved In Perl 6



Jonathan Worthington
Seoul.pm

Solved in Perl 6

About Me

Solved in Perl 6

From the North of England...



Solved in Perl 6

From the North of England...



Solved in Perl 6

From the North of England...



Solved in Perl 6

From the North of England...



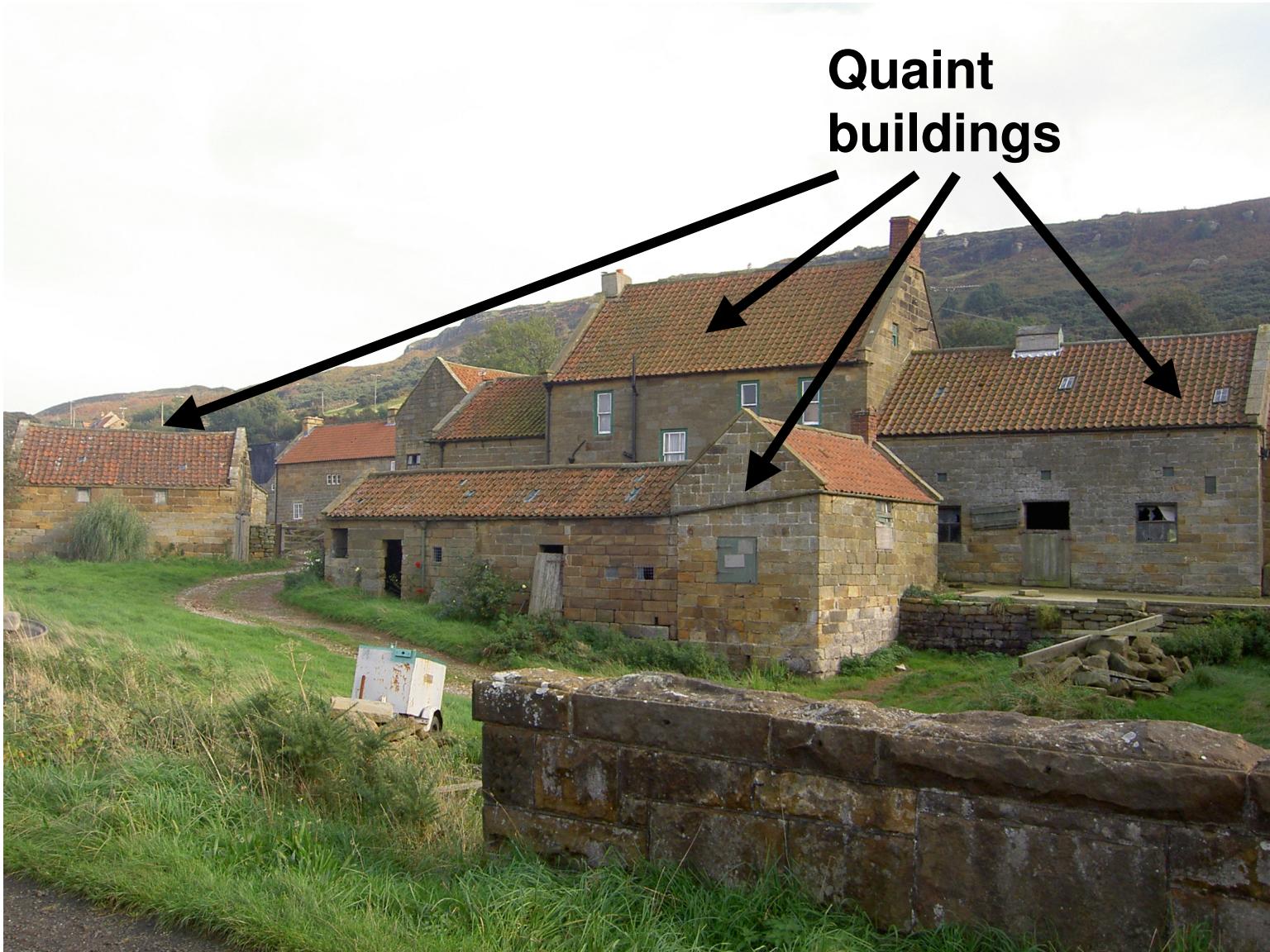
Solved in Perl 6

What North England Looks Like



Solved in Perl 6

What North England Looks Like

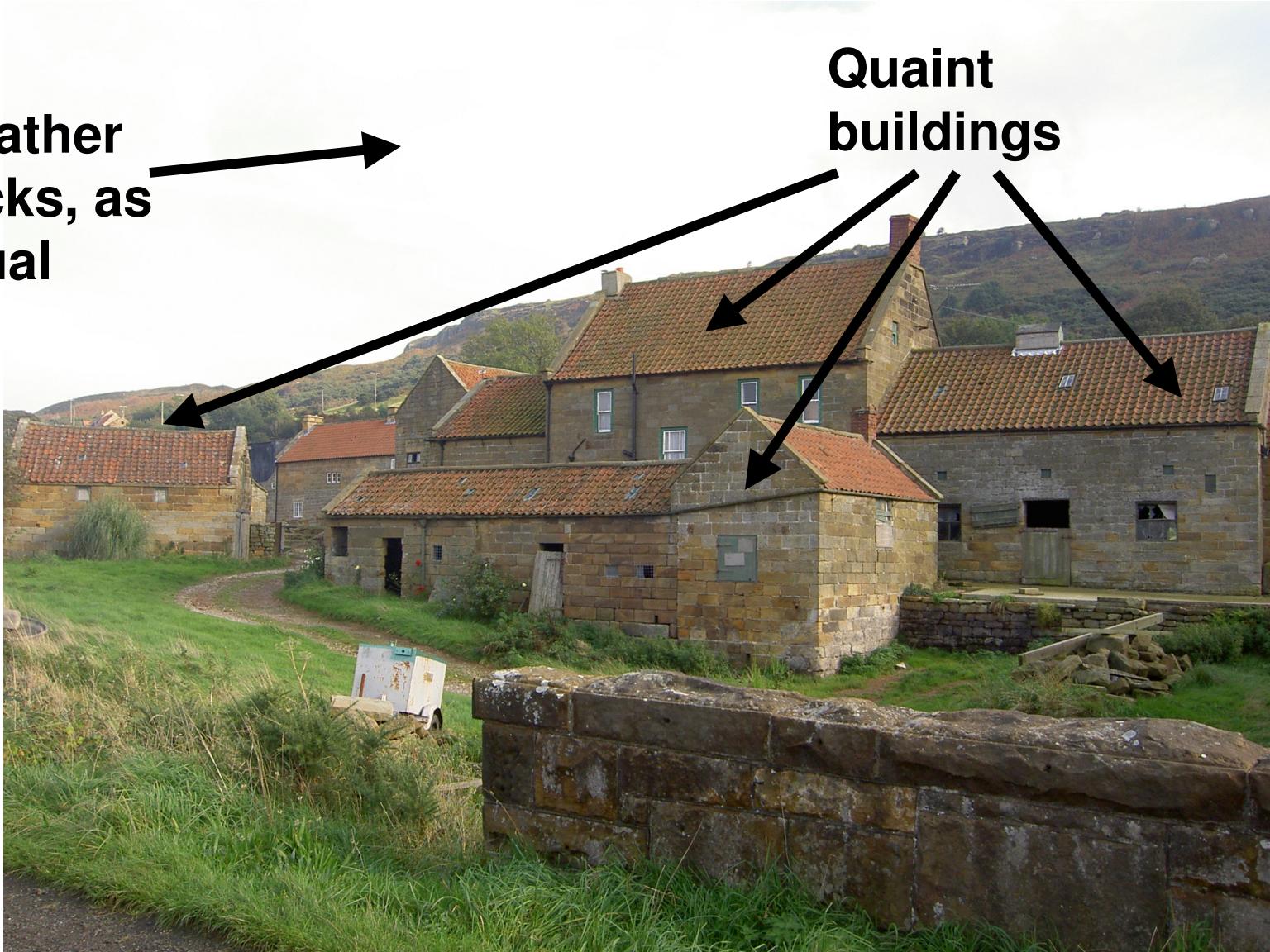


Solved in Perl 6

What North England Looks Like

Weather
sucks, as
usual

Quaint
buildings



Solved in Perl 6

Now I Live In Slovakia



Solved in Perl 6

Now I Live In Slovakia



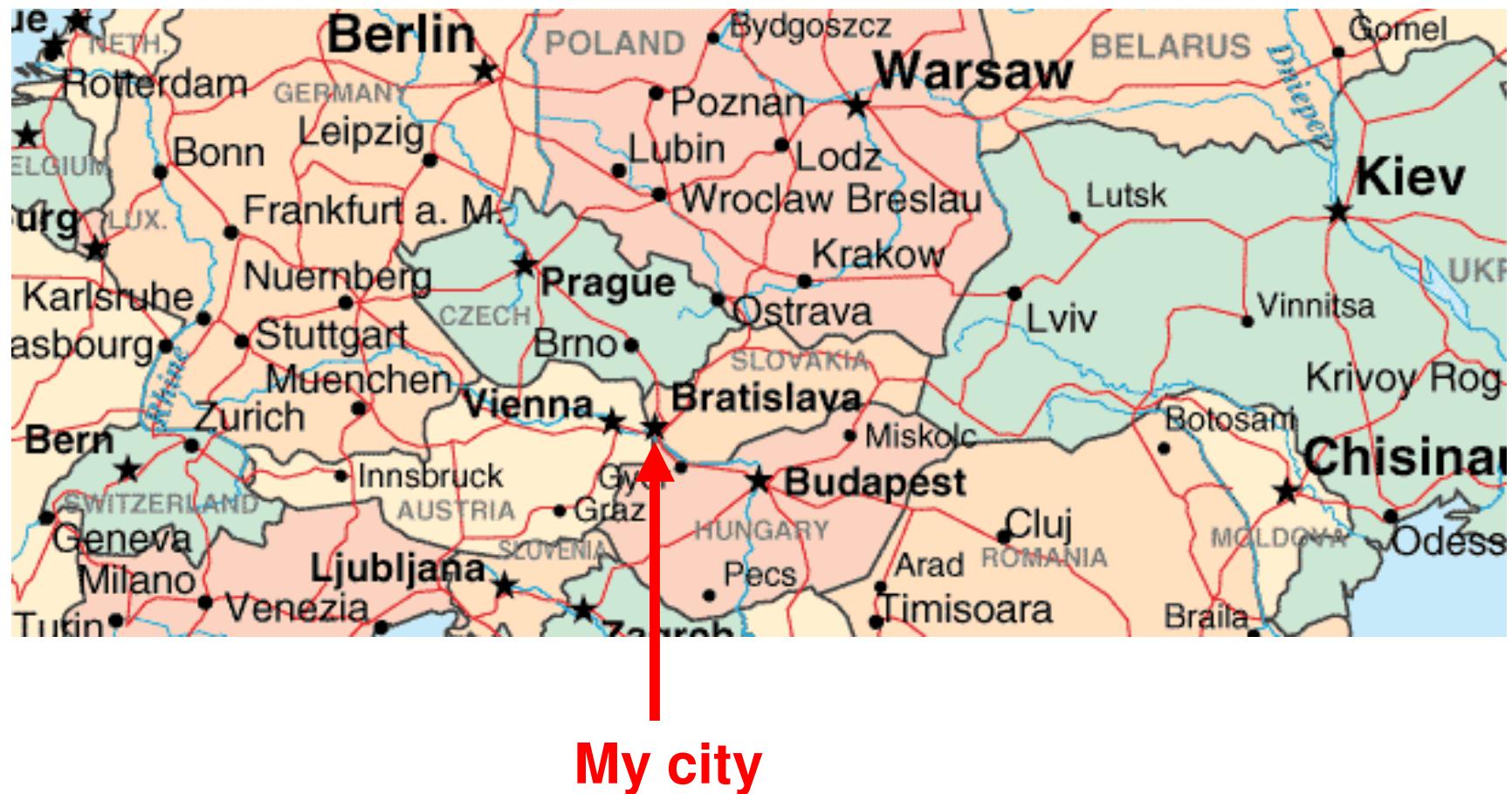
Solved in Perl 6

Now I Live In Slovakia



Solved in Perl 6

Now I Live In Slovakia



Solved in Perl 6

Now I Live In Slovakia



Solved in Perl 6

Now I Live In Slovakia



Solved in Perl 6

Now I Live In Slovakia

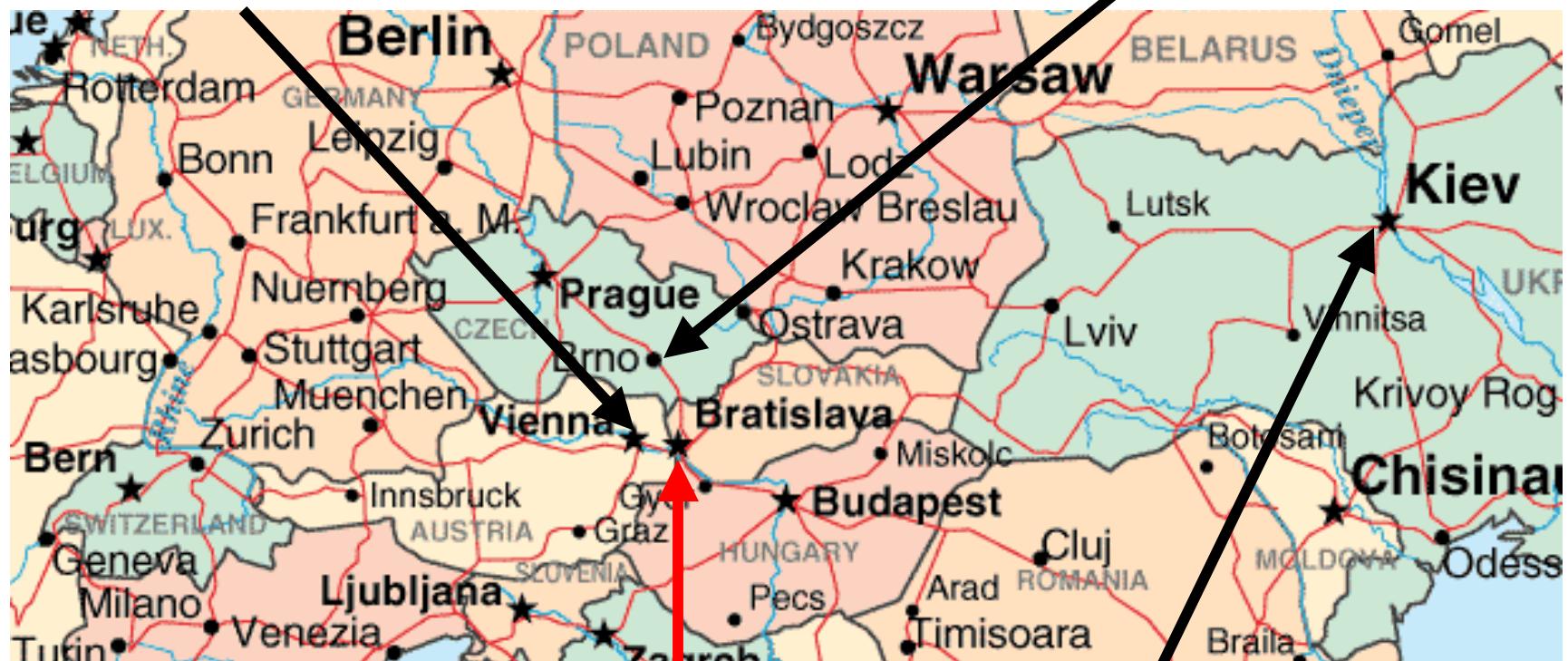


Solved in Perl 6

What's Around?

Source of good
coffee and cake

Source of
good beer



My city

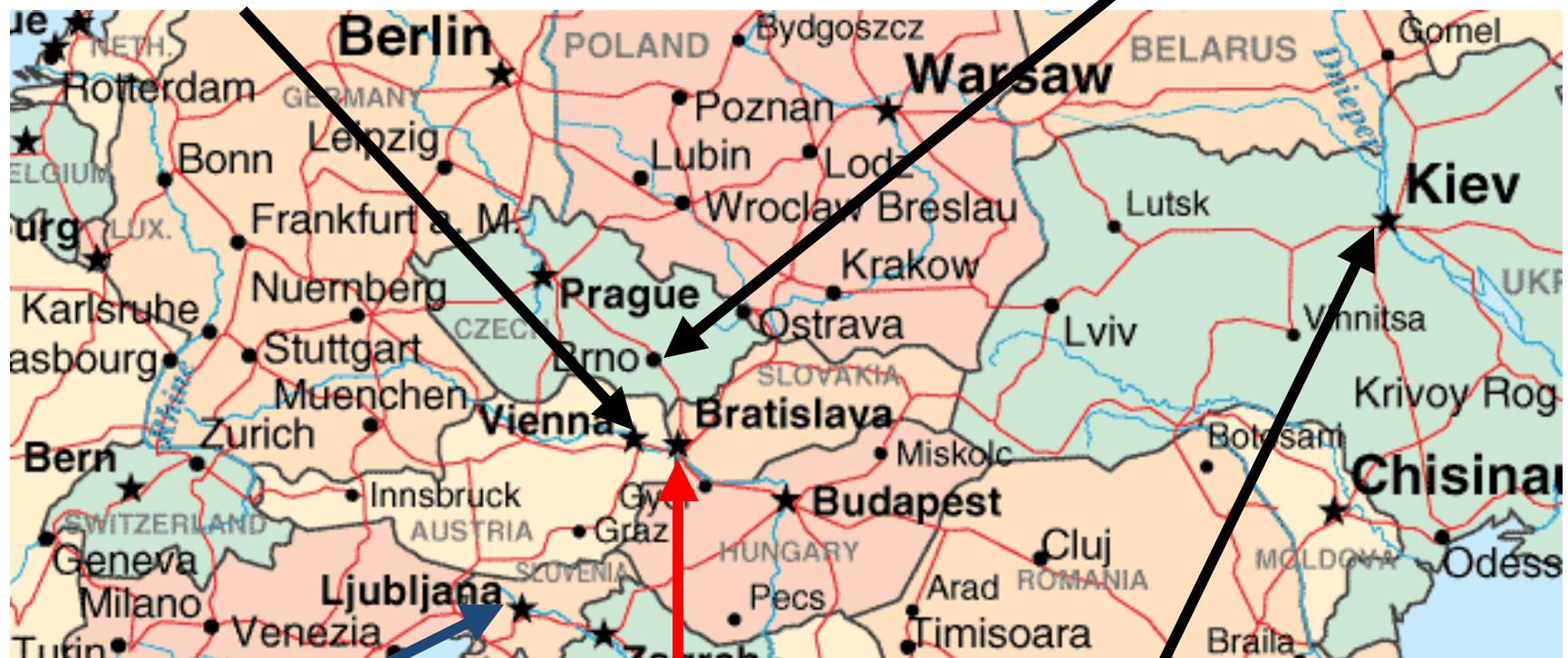
Has a beautiful
prime minister

Solved in Perl 6

What's Around?

Source of good coffee and cake

Source of good beer



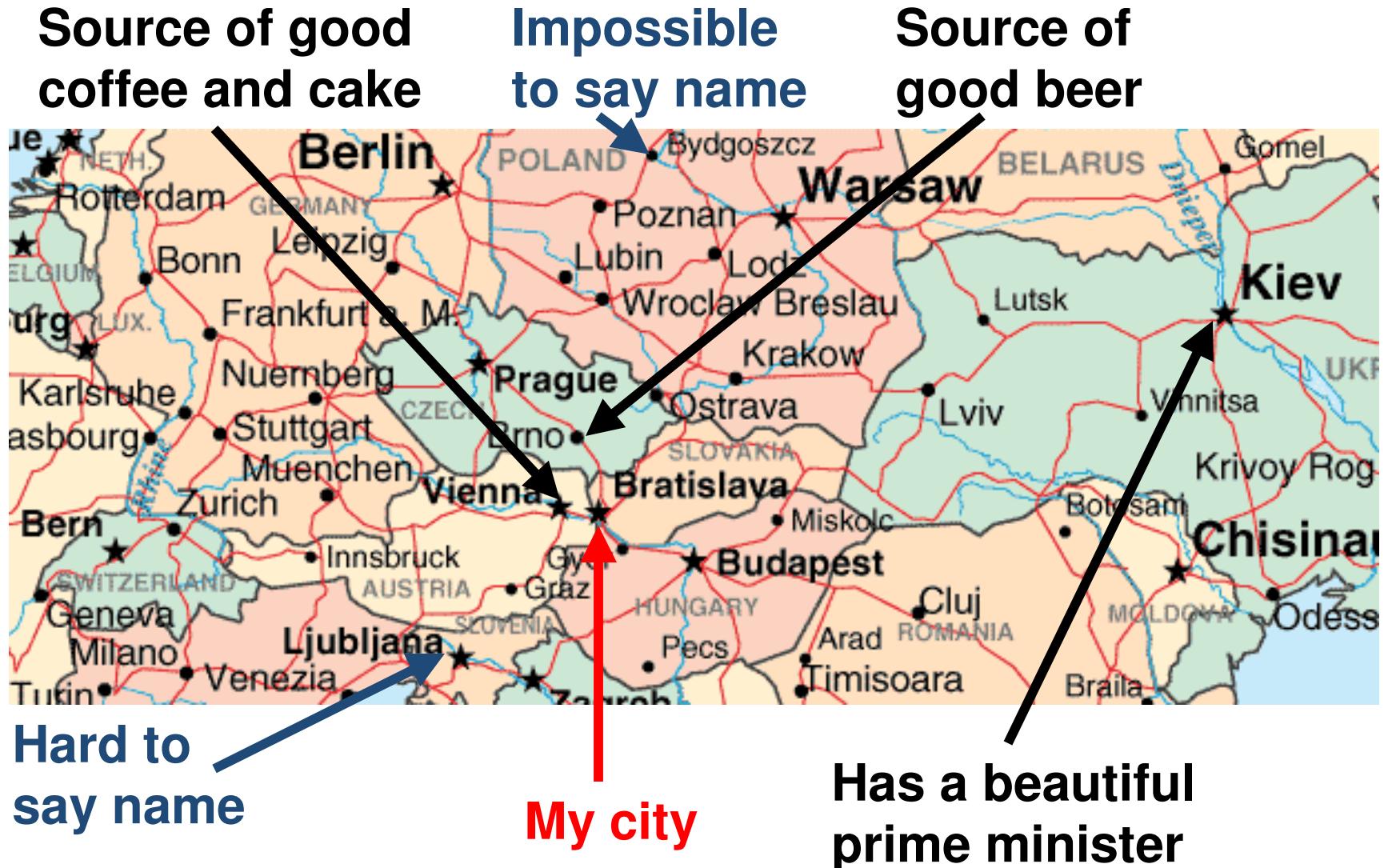
Hard to say name

My city

Has a beautiful prime minister

Solved in Perl 6

What's Around?



Solved in Perl 6

I Like Beer

I know how to say beer
in more languages than
any other word.

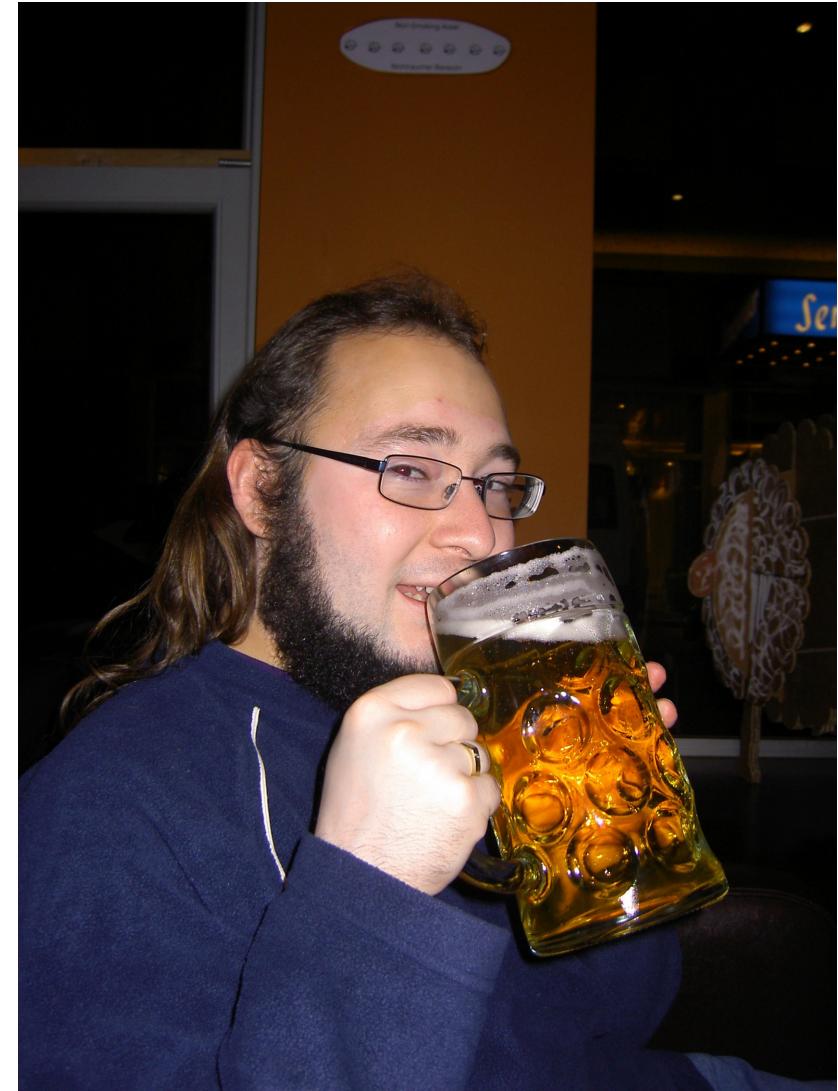
beer / bier /

bière / cerveza /

piwo / pivo /

birra / Пиво /

ビール / 맥주 / ...



Solved in Perl 6



I Love To Travel

- So far, I've visited 32 countries spread over four continents



Solved in Perl 6

I Hack On Perl 6

- Contributed to the Parrot virtual machine since 2003
- Worked on Rakudo Perl 6 compiler since 2007
- Currently one of the lead Rakudo developers
- Specialize in object orientation, type system and multiple dispatch
- Working on Perl 5 <-> Perl 6 integration to aid migration

Solved in Perl 6



My Talk

Solved in Perl 6

What is this talk about?

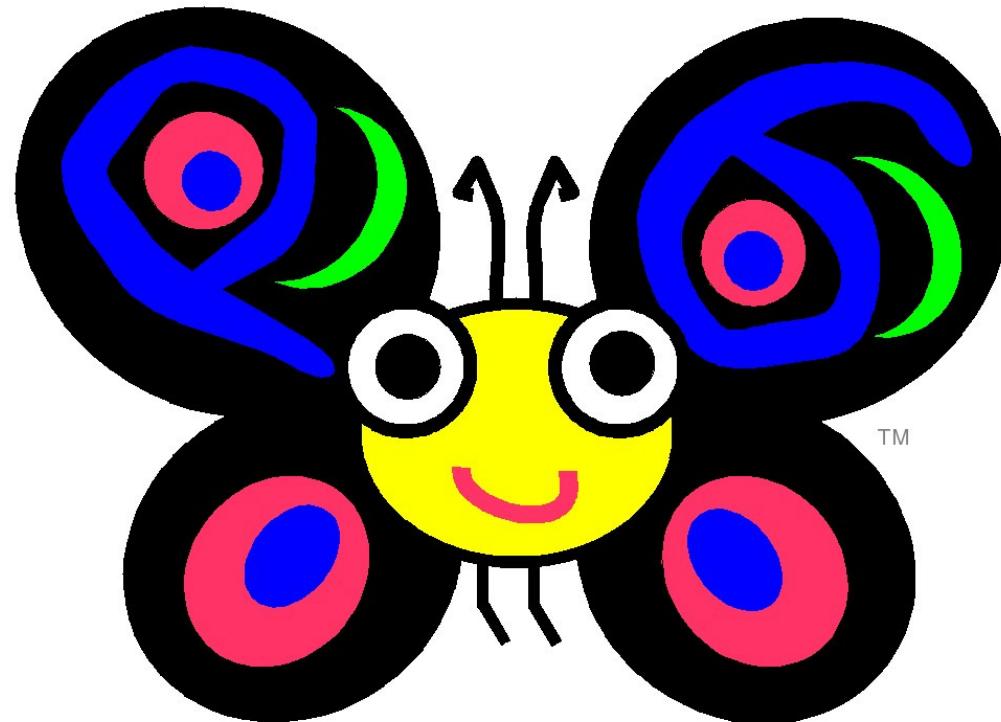
- A little background on the Perl 6 project
- How to do a range of everyday tasks using Perl 6 (all code works today in Rakudo)
- Rakudo release plan – what, when, etc.

본 발표에서 다룰 것은 무엇인가?

- 펄6 프로젝트 이해를 위한 약간의 백그라운드
- 펄6를 이용하여 어떻게 여러 범위의 임무를 수행하는지 (오늘 선보이는 코드는 라쿠도에서 동작 확인)
- 라쿠도 릴리즈 계획 - 무엇을, 언제, 그외.

Solved in Perl 6

Perl 6



Solved in Perl 6

What is Perl 6?

- Re-design and update of the Perl language
- Not syntactically backwards compatible
- Many changes, many new features, but aims to still "feel like Perl"

펄6란 무엇인가?

- 펄언어에 대한 새로운 디자인과 갱신
- 문법상으로 하위 호환성을 제공하지 않는다
- 많은 변경과 수많은 기능들이 있지만 "펄 느낌"을 보존하게끔 노력중

Solved in Perl 6

Specification

- Perl 6 is not an implementation, but rather a language specification
 - Part of it is a written specification
 - Part of it is a "specification test suite"

사양

- 펄6는 구현체라기 보다는 언어에 대한 사양이다
- 일부는 서술된 사양이다
- 일부는 "사양에 대한 test suite"

Solved in Perl 6

Whirlpool Development

- Perl 6 isn't doing the waterfall model, but instead the "whirlpool"
- Feedback from implementers and users helps refine the language specification

소용돌이 모델 개발 프로세스

- 펄6는 폭포수 모델을 따르지 않는다 대신 "소용돌이" 모델이다.
- 개발자들과 사용자들이 제공하는 피드백의 도움으로 언어 사양을 개량한다.

Solved in Perl 6



Implementations

- Pugs was the first serious implementation, but sadly is no longer active
- Active implementations today include Rakudo, smop, Elf and mp6

구현체들

- Pugs는 처음으로 펄6를 진지하게 구현한것이었지만 아쉽게도 더이상 활동이 이루어지지 않음.
- 현재 활동이 이루어지는 구현체들은 라쿠도, smop, Elf 그리고 mp6

Solved in Perl 6



Rakudo Perl[™]

Solved in Perl 6

What is Rakudo?

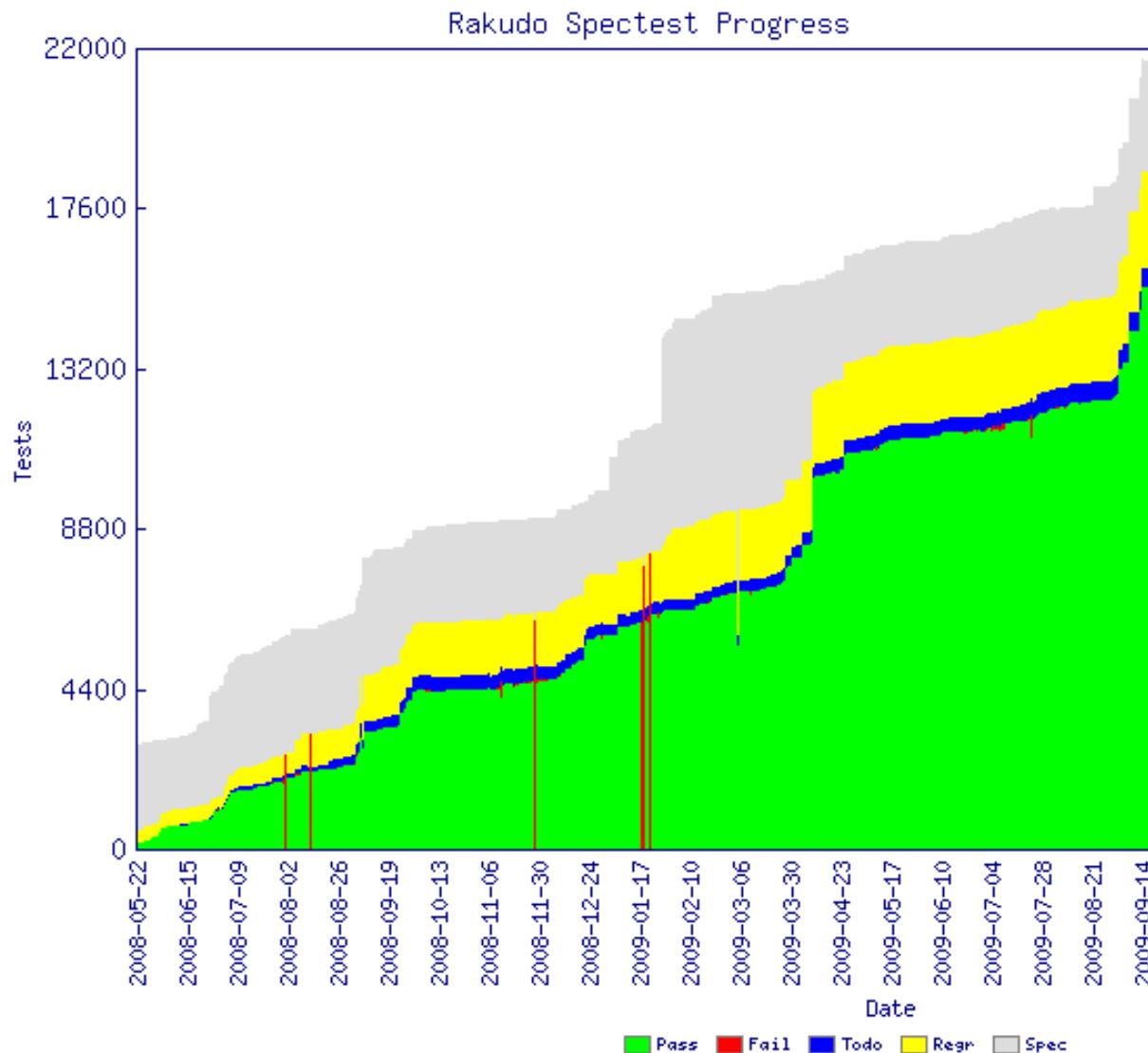
- Implementation of the Perl 6 specification
- Targets the Parrot Virtual Machine
- Currently the most actively developed Perl 6 implementation

라쿠도란 무엇인가?

- 펄6 사양에 대한 구현체
- 패럿 가상 머신을 겨냥한 구현체
- 현재 가장 활발히 개발이 이루어진 펄6 구현체

Solved in Perl 6

Rakudo Progress



15,498
passing
tests!

Solved in Perl 6



Rakudo Progress

- Rakudo supports a wide range of features from the Perl 6 language specification
- All example code I will show today already runs in Rakudo

라쿠도 진행상황

- 라쿠도는 펄6 언어 사양에 기재된 많은 기능들을 지원한다
- 오늘 선보일 코드들은 이미 라쿠도에서 동작함

Solved in Perl 6

Problems & Solutions

Solved in Perl 6



Problem

Say "Hello, world"

"Hello, world"를 출력하기

Solution

```
say "Hello, world!"
```

Output

```
Hello, world!
```

Solved in Perl 6



Problem

Read input from the console

콘솔에서 입력을 받아 들이기

Solution

```
print "Enter your name: ";
my $name = $*IN.get;
say "안녕하세요 $name!";
```

Output

```
Enter your name: Jonathan
```

```
안녕하세요 Jonathan!
```

Solved in Perl 6



Problem

Check a value is in a given range

값이 지정된 범위에 있는지 확인하기

Solution 1

```
print "Enter a number between 1 and 10: ";
my $number = $*IN.get;
unless 1 <= $number <= 10 { say "Oh no!" }
```

Output

```
Enter a number between 1 and 10: 3
```

```
Enter a number between 1 and 10: 42
```

```
Oh no!
```



Problem

Add up a list of numbers

리스트의 합계 값을 구하기

Solution

```
my @nums = 1, 5, 7, -2, 3, 9, 11, -6, 14;  
say [+] @nums;
```

Output

42

Solved in Perl 6



Problem

Check if a list is sorted

리스트가 정렬이 되었는지에 대한 확인

Solution

```
my @a = 1, 1, 2, 3, 5, 8;
my @b = 9, 4, 1, 16, 36, 25;
if [≤] @a { say '@a is sorted' }
if [≤] @b { say '@b is sorted' }
```

Output

```
@a is sorted
```

Solved in Perl 6



Problem

Get a Perl-ish representation of a data structure
펄스러운 데이터구조를 표기하기

Solution

```
my @a = 1, 2, 3;
push @a, { x => 42, y => 100 };
say @a.perl;
```

Output

```
[1, 2, 3, { "y" => 100, "x" => 42 } ]
```

Solved in Perl 6



Problem

Iterate over a list

리스트를 이터레이트 하기

Solution

```
my @cities = <Busan Gyeong-ju Seoul>;
for @cities -> $city {
    say "I've visited $city";
}
```

Output

```
I'll visited Busan
I'll visited Gyeong-ju
I'll visited Seoul
```

Solved in Perl 6



Problem

Iterate over the keys and values of a hash
해쉬의 키와 해당 값으로 이터레이트 하기

Solution

```
my %distances = Busan => 329, Daegu => 236;
for %distances.kv -> $city, $distance {
    say "$city is $distance km away";
}
```

Output

```
Busan is 329 km away
Daegu is 236 km away
```

Solved in Perl 6



Problem

Check if any of a list of test scores is a pass

리스트 안에 있는 아이템중 해당 테스트를 통과 하는지에
대한 확인

Solution

```
my @a = 75, 47, 90, 22, 80;
my @b = 61, 77, 94, 82, 60;
my @c = 45, 59, 33, 11, 19;
if any(@a) >= 60 { say "Some passes in A" }
if any(@b) >= 60 { say "Some passes in B" }
if any(@c) >= 60 { say "Some passes in C" }
```

Output

```
Some passes in A
Some passes in B
```

Solved in Perl 6



Problem

Check if all of a list of test scores are passes

리스트 안에 있는 모든 아이템이 해당 테스트를 통과 하는지에 대한 확인

Solution

```
my @a = 75, 47, 90, 22, 80;
my @b = 61, 77, 94, 82, 60;
my @c = 45, 59, 33, 11, 19;
if all(@a) >= 60 { say "All passes in A" }
if all(@b) >= 60 { say "All passes in B" }
if all(@c) >= 60 { say "All passes in C" }
```

Output

```
All passes in B
```

Solved in Perl 6



Problem

Check if none of a list of test scores is a pass

리스트 안에 있는 모든 아이템이 해당 테스트를
통과 못하는지에 대한 확인

Solution

```
my @a = 75, 47, 90, 22, 80;
my @b = 61, 77, 94, 82, 60;
my @c = 45, 59, 33, 11, 19;
if none(@a) >= 60 { say "No passes in A" }
if none(@b) >= 60 { say "No passes in B" }
if none(@c) >= 60 { say "No passes in C" }
```

Output

No passes in C

Solved in Perl 6



Problem

Get a random item from a list

리스트에서 랜덤으로 아이템 받기

Solution

```
my @drinks = <soju beer vodka>;  
say "Tonight I'll drink { @drinks.pick }";
```

Output (results should vary ;-))

```
Tonight I'll drink soju
```

Solved in Perl 6



Problem

Shuffle a list into a random order

리스트를 셔플해서 랜덤으로 순서를 받기

Solution

```
my @competitors = <Tina Lena Owen Peter>;
my @order = @competitors.pick(*);
@order>>.say;
```

Output (results should vary ;-))

```
Peter
Lena
Owen
Tina
```

Solved in Perl 6



Problem

Write and call a subroutine with parameters
파라미터로 서브루틴을 작성하고 호출하기

Solution

```
sub greet ($name) {  
    say "안녕하세요 $name!";  
}  
greet ("Patrick");
```

Output

안녕하세요 Patrick!

Solved in Perl 6



Problem

Write a subroutine taking an array and a hash
배열과 해쉬를 인자로 받는 서브루틴 작성하기

Solution

```
sub example(@a, %h) {  
    say @a.elems;  
    say %h.keys;  
}  
my @nums = 42, 57, 74;  
my %mapping = a => 1, b => 2;  
example(@nums, %mapping);
```

Output

```
3  
ab
```



Problem

Write a subroutine that only takes a number
숫자를 인자로만 받는 서브루틴 작성하기

Solution

```
sub double(Num $n) { 2 * $n }
say double(21);
say double("oh no I'm not a number");
```

Output

42

Parameter type check failed; expected Num,
but got Str for \$n in call to double



Problem

Use multi-subs to react differently by type

형태에 따라서 동작이 다른 다중 서브루틴 작성하기

Solution

```
multi double(Num $n) { 2 * $n }
multi double(Str $s) { $s x 2 }
say double(21);
say double("boo");
```

Output

42

booboo

Solved in Perl 6



Problem

Compute factorial (recursively)
계승을 계산하기 (재귀적으로)

Solution

```
multi fact($n) { $n * fact($n - 1) }
multi fact(0) { 1 }
say fact(1);
say fact(10);
```

Output

```
1
3628800
```

Solved in Perl 6



Problem

Compute factorial (using a meta-operator)
계승을 계산하기 (메타 연산자를 사용하여)

Solution

```
sub fact ($n) { [*] 1..$n }
say fact(1);
say fact(10);
```

Output

```
1
3628800
```



Problem

Add a new factorial operator (so $10!$ works)

새로운 계승연산자를 추가하기 ($10!$ 라고 작성할 수 있도록)

Solution

```
sub postfix:<!>($n) { [*] 1..$n }
say 1!;
say 10!;
```

Output

```
1
3628800
```

Solved in Perl 6

Problem

Declare a class with attributes and a method

에트리뷰트(attribute) 와 메소드를 가지는 클래스의 선언

Solution

```
class Product {  
    has $.name;    # Attr + accessor  
    has $!price;  # Attr only  
    has $.discount is rw;  
                # Attr + lvalue accessor  
    method get_price {  
        return $!price - $!discount;  
    }  
}
```



Problem

Instantiate a class and call a method on it

클래스의 인스턴스화와 메소드의 호출

Solution

```
my $prod = Product.new(
    name      => "Beer",
    price     => 500,
    discount  => 60
);
say $prod.get_price;
```

Output



Problem

Get/set attributes through accessors

엑세서(accessor)를 사용하여 에트리뷰트(attribute)를 획득/설정

Solution

```
say $prod.name;  
$prod.discount = 40;  
say $prod.get_price;  
$prod.name = 'Wine';
```

Output

```
Beer  
460
```

```
Cannot assign to readonly variable.
```

Solved in Perl 6

Problem

Call a method on every object in a list
리스트의 모든 아이템에 메소드 호출

Solution

```
my @products =  
    Product.new(name => 'Beer', price => 500),  
    Product.new(name => 'Wine', price => 450),  
    Product.new(name => 'Vodka', price => 1600);  
for @products>>.name>>.uc { .say }
```

Output

```
BEER  
WINE  
VODKA
```

Solved in Perl 6



Problem

Introspect a class to find its methods

클래스의 내용을 살펴보고 메소드를 찾기

Solution

```
my @meths = Product.^methods(:local);  
for @meths>>.name { .say }
```

Output

```
get_price  
discount  
name
```

Solved in Perl 6

Problem

Sort an array of objects by result of a method

메소드의 결과를 이용해 오브젝트의 배열을 정렬

Solution (Example 1)

```
my @products =  
    Product.new(name => 'Beer', price => 500),  
    Product.new(name => 'Wine', price => 450),  
    Product.new(name => 'Vodka', price => 1600);  
for @products.sort(*.name) >> .name { .say }
```

Output (Example 1)

```
Beer  
Vodka  
Wine
```

Solved in Perl 6

Problem

Sort an array of objects by result of a method

메소드의 결과를 이용해 오브젝트의 배열을 정렬

Solution (Example 2)

```
my @products =  
    Product.new(name => 'Beer', price => 500),  
    Product.new(name => 'Wine', price => 450),  
    Product.new(name => 'Vodka', price => 1600);  
for @products.sort(*.get_price) >> .name { .say }
```

Output (Example 2)

```
Wine  
Beer  
Vodka
```

Solved in Perl 6



Problem

Find minimum and maximum values from a list
리스트로부터 최소치, 최대치를 찾아내기

Solution (Example 1)

```
my @temperatures = -3, 5, 7, 2, -1, -4, 0;  
say "Minimum was " ~ @temperatures.min;  
say "Maximum was " ~ @temperatures.max;
```

Output (Example 1)

```
Minimum was -4  
Maximum was 7
```

Solved in Perl 6



Problem

Find minimum and maximum values from a list
리스트로부터 최소치, 최대치를 찾아내기

Solution (Example 2)

```
my @products =
    Product.new(name => 'Beer', price => 500),
    Product.new(name => 'Wine', price => 450),
    Product.new(name => 'Vodka', price => 1600);
say "Cheapest: " ~ @products.min(*.get_price).name;
say "Costliest: " ~ @products.max(*.get_price).name;
```

Output (Example 2)

```
Cheapest: Wine
Costliest: Vodka
```



Problem

Paper, Scissor, Stone game

가위,바위,보 게임

Solution (Part 1)

```
class Paper { }
class Scissor { }
class Stone { }

multi win(Paper,    Stone)    { "Win"   }
multi win(Scissor,  Paper)    { "Win"   }
multi win(Stone,    Scissor)  { "Win"   }
multi win(::T,       T)       { "Draw"  }
multi win(Any,      Any)      { "Lose" }
```

Solved in Perl 6



Problem

Paper, Scissor, Stone game
가위,바위,보 게임

Solution (Part 2)

```
say win(Paper, Paper);
say win(Scissor, Stone);
say win(Stone, Scissor);
```

Output

```
Draw
Lose
Win
```

Solved in Perl 6



So much
nice stuff....

Solved in Perl 6



...but when
will something
be released?

Solved in Perl 6

The Good News About Rakudo

- Very actively developed
- More than one person understands the guts
- So far, 20 monthly development releases have been made

라쿠도에 대한 좋은 소식

- 현재 매우 활발히 개발중
- 한명 이상이 상세한 부분까지 이해를 했음
- 현재 20개이상의 월별 단위로 개발 릴리스가 제공 되었음

Solved in Perl 6



The Problem

- Rakudo needs applications and modules written for it, to find spec/Rakudo issues
- However, few people will use Rakudo if we keep on making "development releases"

문제점들

- 라쿠도의 사양과 구현체의 문제들을 찾기 위해 프로그램과 모듈이 필요함
- 하지만 계속 "개발 버전" 만 만든다면, 몇몇 사람들만이 라쿠도를 사용할 것이다.

Solved in Perl 6



Therefore...

- In Q2 2010, we will make a major, usable, useful release called Rakudo *
- An implementation of a significant subset of the Perl 6 language specification

그런즉...

- 2010년 2분기에는 라쿠도라는 사용성있고 유용한 메이저 릴리즈를 할것이다
- 펄6 언어 사양의 많은 부분을 커버하는 구현체가 될것이다

Solved in Perl 6



Making Perl 6 Accessible

- Right now, Rakudo is mostly used by those with some interest in Perl 6 development
- Rakudo * aims to make it more accessible
- It's a little like people visiting a mountain...

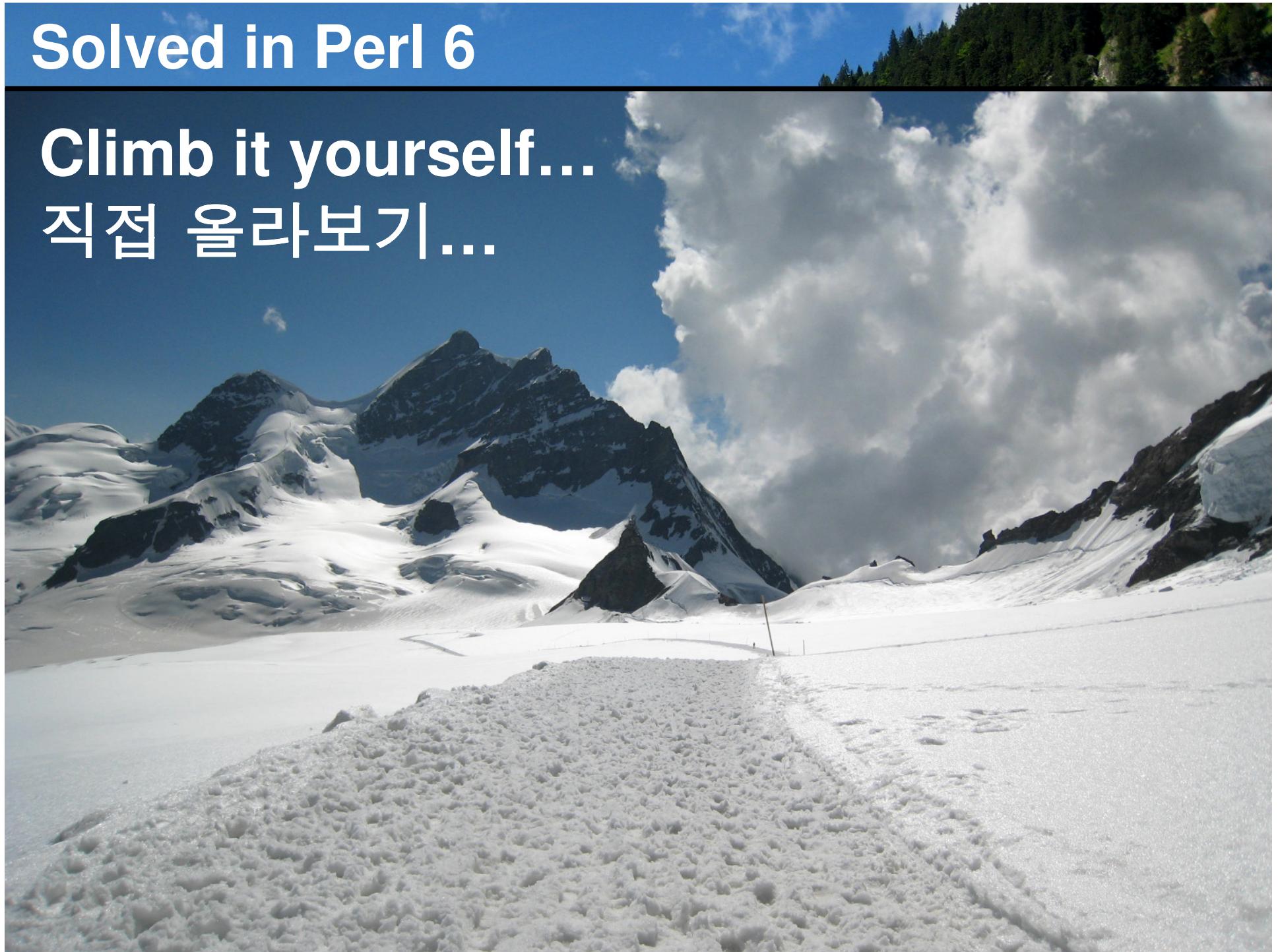
펄6를 더 많은 사람들이 접하게끔 하기

- 라쿠도는 대부분 펄 6 개발에 관심있는 사람들만 사용하고 있다
- 라쿠도는 더 많은 사람들이 펄6를 접근할수 있도록 목표하고 있다
- 사람들이 등산하러 오는것과 흡사하다...

Solved in Perl 6

Climb it yourself...

직접 올라보기...



Solved in Perl 6



Climb It Yourself

- Only appealing to a relatively small group of people
- Risky but, when it works out, rewarding
- This is where the Rakudo compiler is today

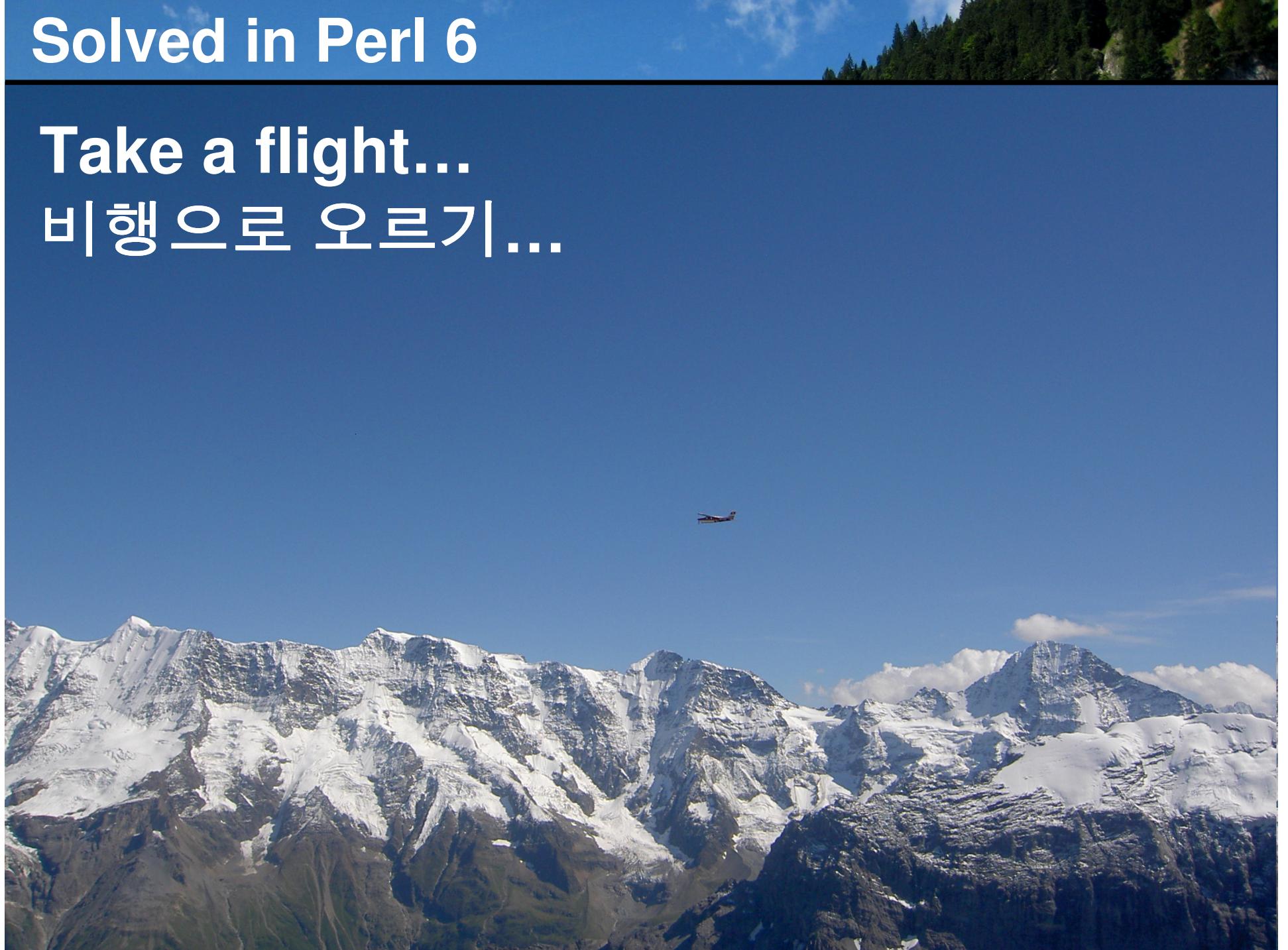
직접 올라보기

- 소규모의 그룹으로 구성된 사람들한테만 매력적이다
- 모험적이긴 하지만 제대로 진행이 되면 상당한 보답이 있을것이다
- 이를 가능케 해주는 예가 바로 라쿠도 컴파일러이다

Solved in Perl 6

Take a flight...

비행으로 오르기...





Take A Flight

- Much easier = many more people can do it
- You can enjoy the mountain...but you know that standing on it would be better still
- This is where Rakudo * is aiming at

비행으로 오르기

- 훨씬 쉬움 = 많은 사람들이 할수 있음
- 산을 보는것을 즐길수도 있지만... 정상에 서 보는것이 훨 나을것이다
- 이것이 바로 라쿠도의 목표!

Solved in Perl 6

The train to the top

산정상을 위한 기차





The Train To The Top

- Very accessible – even to those with a fear of flying (e.g. early adoption)
- You're on the mountain
- But it takes a lot of time and effort to build

산정상을 위한 기차

- 매우 접근하기 용이하다 - 나는게 무섭다는 사람들 한테도 (e.g. 조기 도입)
- 이미 산을 등반하고 있는것이다
- 허나 이를 구축하기 위해선 많은 시간과 노력이 든다

Solved in Perl 6



A Step On The Journey

- Rakudo * isn't all of Perl 6, but instead a significant step towards it
- Enough of Perl 6 to be useful to many
- But we know it won't be for everyone

여행을 위한 첫걸음

- 라쿠도는 펄6의 모든 부분을 목표하진 않지만 그러기 위한 큰 한 걸음
- 많은 사람들에게 유용할수 있는 충분한 펄6가 되기
- 그러나 모든 사람들을 만족시키지 못한 다는 것을 알고 있다

Solved in Perl 6

Want to help us along
the path to Rakudo *?



Solved in Perl 6

How To Get Involved

- Write modules and applications; report the bugs / frustrations you encounter
- The spec test suite always needs work
- www.perl6.org / [#perl6](irc://irc.freenode.org)

어떻게 참여 할수 있는가

- 모듈과 프로그램을 작성하기; 버그와 불만을 보고하기
- 공식사양 test suite는 항상 손질이 필요하다
- www.perl6.org / [#perl6](irc://irc.freenode.org)

Solved in Perl 6



감사합니다

Solved in Perl 6



Questions?

Solved in Perl 6

Rakudo *
Q2 2010
www.rakudo.org

