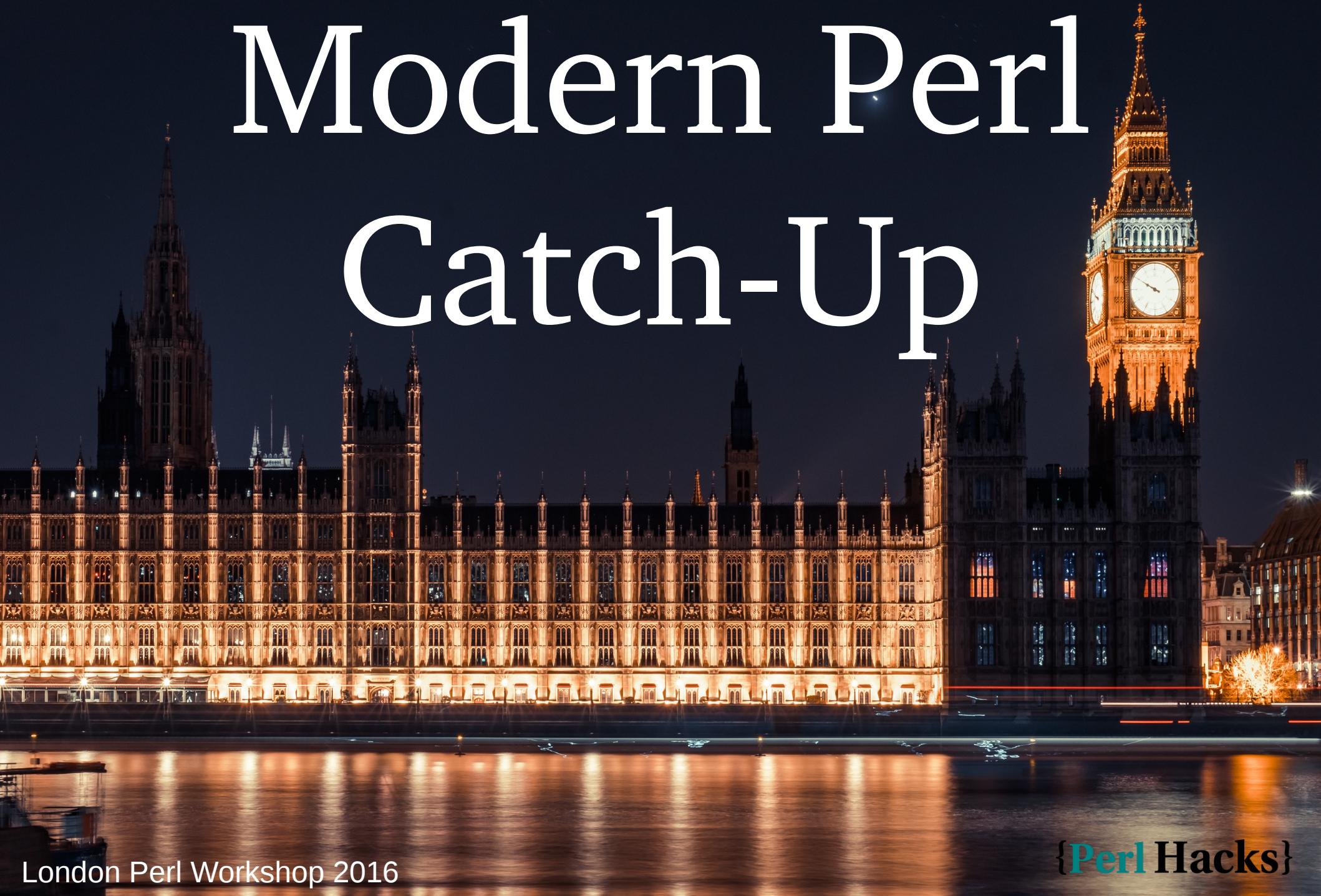


Modern Perl Catch-Up



Modern Perl Catch-Up

- New things in Perl that you might have missed
- Haven't we done this before?
- "Modern Core Perl" December 2011
- This is a sequel

Perl Releases

- Perl is released annually
- In the spring
- Even-numbered versions are production releases
 - 5.22, 5.24, 5.26
- Odd-numbered versions are development releases
 - 5.21, 5.23, 5.25
- 5.25 will become 5.26

Perl Support

- Perl releases are officially supported for two years
- `perldoc perlpolicy`
- Currently 5.22 and 5.24
- Limited support for older versions
- Longer support from vendors

Update Your Perl

- Using the system Perl is a bad idea
- Often out of date
- perlbrew and plenv

5.16



5.16

- 5.16.0 - 20 May 2012
- 5.16.1 - 08 Aug 2012
- 5.16.2 - 01 Nov 2012
- 5.16.3 - 11 Mar 2013

5.16 Features

- SUB
- Fold case
- Devel::DProf removed
- perlexperiment
- perlootut
- Old OO docs removed
- Unicode 6.1

SUB

- New special source code token
- Like `__PACKAGE__`, `__FILE__` and `__LINE__`
- Returns a reference to the current subroutine
- Easier to write recursive subroutines

Recursive without SUB

```
sub fib {  
    my ($n) = @_;  
  
    return 1 if $n <= 2;  
    return fib($n - 1) + fib($n - 2);  
}
```

Recursive with __SUB__

```
sub fib {  
    my ($n) = @_;  
  
    return 1 if $n <= 2;  
    return __SUB__->($n - 1) +  
          __SUB__->($n - 2);  
}
```

Recursive with __SUB__

```
sub fib {  
    my ($n) = @_;  
  
    return 1 if $n <= 2;  
    return __SUB__->($n - 1) +  
          __SUB__->($n - 2);  
}
```

Fold Case

- `lc($this) eq lc($that)`
- Unicode complicates everything
- Lower-case comparisons can fail
- As can upper-case comparisons
- "Fold-case" is guaranteed to work
 - `fc($this) eq fc($that)`
 - Also `\F` (like `\L` and `\U`)

Devel::DProf Removed

- Devel::DProf was an old profiling tool
- Few people know how to use it
- We're better off without it
- Devel::NYTProf is much better

perldoc perlexperiment

- New documentation page
- Lists the experiments in the current version of Perl
 - Vital to use the right version
- Compare perldoc experimental

perldoc perlootut

- New OO tutorial
- Improvement on old version
- Up to date
- Covers OO frameworks
 - Moose, Moo, Class::Accessor, Class::Tiny, Role::Tiny
- So consider those the "approved" list of OO frameworks

Old OO docs removed

- Outdated OO documentation removed
- perlboot (basic OO tutorial)
- perltoot (Tom's OO tutorial)
- perlbot (bag of object tricks)
- perlootut replaces all of these

Unicode 6.1

- Unicode complicates everything!
- <http://www.unicode.org/versions/Unicode6.1.0>
- "This latest version adds characters to support additional languages of China, other Asian countries, and Africa. It also addresses educational needs in the Arabic-speaking world. A total of 732 new characters have been added."

5.18



5.18

- 5.18.0 - 18 May 2013
- 5.18.1 - 12 Aug 2013
- 5.18.2 - 06 Jan 2013
- 5.18.3 - 01 Oct 2014
- 5.18.4 - 01 Oct 2014

5.18 Features

- no warnings experimental::foo
- Hash rework
- Lexical subroutines
- Smartmatch is experimental
- Lexical `$_` is experimental
- `$/ = \$/n` reads `$n` characters (not bytes)
- `qw(. . .)` needs extra parentheses
- Unicode 6.2

no warnings experimental::foo

- New "warnings" category - experimental
- Warns if you use experimental features
- no warnings experimental::foo
- "Don't warn me that I'm using experimental feature 'foo'"
- "I know what I'm doing here!"

Hash Rework

- Big change in the way hashes are implemented
- Hash ordering is even more random
- Random seed in hashing algorithm
- keys (and values, etc) returns different order each run
- Algorithmic complexity attacks

Example

```
$ perl -E'%h=(one=>1,two=>2,three=>3); say keys %h'  
onethreetwo  
$ perl -E'%h=(one=>1,two=>2,three=>3); say keys %h'  
threetwoone  
$ perl -E'%h=(one=>1,two=>2,three=>3); say keys %h'  
threetwoone  
$ perl -E'%h=(one=>1,two=>2,three=>3); say keys %h'  
threeonetwo
```

Previously

- With earlier versions, the same hash would return the keys in the same order each time
- Now you can't rely on that
- Exposes weaknesses in tests
- Careful testing required!

Lexical Subroutines

- Experimental feature
- no warnings experimental::lexical_subs
- Truly private subroutines
- Only visible within your lexical scope
- state sub foo { ... }
- my sub bar { ... }

state vs my

- Similar to variable declarations
- `my` subroutines are recreated each time their scope is executed
- `state` subroutines are created once
- `state` subroutines are slightly faster
- `my` subroutines are required for closures
- Also `our` for access to a global subroutine of the same name

Private Subroutines

- We name private subroutines with leading `_`
- `sub _dont_run_this { ... }`
- "Please don't run this"
- We can now enforce this
- `state sub _cant_run_this { ... }`
- Only visible within current lexical scope

Smartmatch is Experimental

- Smartmatch is confusing
- Smartmatch has changed a few times
- Smartmatch should probably be avoided
- Smartmatch is experimental
- no warnings
`experimental::smartmatch`
- Warns on `~~`, `given` and `when`

Smartmatch

“Smart match, added in v5.10.0 and significantly revised in v5.10.1, has been a regular point of complaint. Although there are a number of ways in which it is useful, it has also proven problematic and confusing for both users and implementors of Perl. There have been a number of proposals on how to best address the problem. It is clear that smartmatch is almost certainly either going to change or go away in the future. Relying on its current behavior is not recommended.”

Lexical `$_` is Experimental

- Introduced in 5.10
- "it has caused much confusion with no obvious solution"
- no warnings
`experimental::lexical_topic`
- Spoilers: Removed in 5.24

`$ / = \$_n` Reads `\$_n` Characters

- Setting `$ /` to a reference to an integer
- (Previously) Reads that number of bytes
- (Now) Reads that number of characters
- This is almost certainly what you want
- Unicode complicates everything

qw(. . .) Needs Extra Parentheses

- Correcting a parsing bug
- `foreach qw(foo bar baz) { . . . }`
- Perl previously added the missing parentheses
- Now it doesn't
- `foreach (qw(foo bar baz)) { . . . }`

Unicode 6.2

- Unicode complicates everything!
- <http://www.unicode.org/versions/Unicode6.2.0>
- "Version 6.2 of the Unicode Standard is a special release dedicated to the early publication of the newly encoded Turkish lira sign."

5.20



5.20

- 5.20.0 - 27 May 2014
- 5.20.1 - 14 Sep 2014
- 5.20.2 - 14 Feb 2015
- 5.20.3 - 12 Sep 2015

5.20 Features

- Subroutine signatures (experimental)
- `%hash{...}` and `%array[...]` slices
- Postfix dereferencing (experimental)
- Unicode 6.3

Subroutine Signatures

- Subroutines can now have a signature
- `sub foo ($bar, $baz) { ... }`
- `use feature 'signatures'`
- Experimental
- no warnings
`experimental::signatures`

Mandatory Parameters

- `sub foo ($bar, $baz) { ... }`
- Checks the number of parameters
- Throws exception if the wrong number
- Assigns parameters to `$bar` and `$baz`

Unnamed Parameter

- Parameters can be unnamed
- Use \$ without a name
- `sub foo ($bar, $, $baz) { ... }`
- Must pass three parameters
- Second parameter is not assigned to a variable
- All parameters still in @_

Optional Parameters

- Make parameters optional by giving them a default
- `sub foo ($bar, $baz = 0) { ... }`
- Call with one or two parameters
- `$baz` set to 0 by default

Slurpy Parameters

- 'Slurpy' parameters follow postional parameters
- Assign them to an array
 - `sub foo ($bar, @baz) { ... }`
 - `foo('a', 'b', 'c');`
- Or a hash
 - `sub foo ($bar, %baz) { ... }`
 - `foo('a', 'b', 'c');`
 - `foo('a', b => 'c');`

`%hash{...}` and `%array[...]` Slices

- Old array slices
- `@slice = @array[1, 3..5]`
- Returns four elements from the array
- Also with hashes
- `@slice = @hash{ 'foo', 'bar', 'baz' }`
- Returns three values from the hash

New Hash Slices

- New `%h{...}` slice syntax
- `%subhash = %hash{ 'foo', 'bar', 'baz' }`
- Returns three key/value pairs
- Also with arrays
- `@slice = %array[1, 3..5]`
- Returns four index/element pairs

Postfix Dereferencing

- no warnings experimental::postderef
- New syntax for dereferencing
- Follows left-to-right rules throughout

Postfix vs Prefix

- `$sref->$*;` # same as `$($sref)`
- `$aref->@*;` # same as `@{ $aref }`
- `$href->%*;` # same as `%{ $href }`
- `$cref->&*;` # same as `&{ $cref }`
- `$gref->**;` # same as `*{ $gref }`

Left to Right Throughout

- Old style
- `@arr = @{$r->{foo}[0]{bar}}}`
- New style
- `@arr = $r->{foo}[0]{bar}->@*`

Unicode 6.3

- Unicode complicates everything!
- <http://www.unicode.org/versions/Unicode6.3.0>
- "Version 6.3 of the Unicode Standard is a special release focused on delivering significantly improved bidirectional behavior."

5.22



5.22

- 5.22.0 - 01 Jun 2015
- 5.22.1 - 13 Dec 2015
- 5.22.2 - 29 Apr 2016

5.22 Features

- New bitwise operators
- Double diamond operator
- New regex boundaries \b{ . . . }
- New /n option on regexes
- Unicode 7.0
- Omitting % and & on hash and array names now an error

5.22 Features (cont)

- `defined(@array)` and `defined(%hash)` now fatal error
- `%hash->{$key}` and `@array->[$idx]` now fatal errors
- perlunicook
- find2perl, a2p, s2p removed
- CGI removed
- Module::Build removed

New Bitwise Operators

- Old bitwise operators
- `~, &, |, ^`
- Expect numbers as operands
- "Work" on strings too
- No numeric/string differentiation

New Bitwise Operators

- use feature 'bitwise';
- no warnings
'experimental::bitwise';
- Or use experimental 'bitwise';
- Old operators are always numeric
- New string bitwise operators
- ~ ., & ., | ., ^ .

Double Diamond Operator

- <<>>
- Like <>
- Uses three-arg open()
- All elements of @ARGV treated as filenames
- | foo will no longer work

New Regex Boundaries

- Extensions to \b
- \b{wb} - Word boundary
 - Like \b but cleverer about punctuation within words
- \b{sb} - Sentence boundary
- \b{gcb} - Grapheme cluster boundary
 - Unicode complicates everything!

/n Option on Regexes

- New option for `m/. . . /` and `s/. . . / . . . /`
- Turns off call capturing
- `/(\w+)/ -` fills in `$1`
- `/(\w+)/n -` doesn't
- See also `(?:\w+)`

Unicode 7.0

- Unicode complicates everything!
- <http://www.unicode.org/versions/Unicode7.0.0>
- "Unicode 7.0 adds a total of 2,834 characters, encompassing 23 new scripts and many new symbols, as well as character additions to many existing scripts"

Unicode 7.0

- Two newly adopted currency symbols: the manat, used in Azerbaijan, and the ruble, used in Russia and other countries
- Pictographic symbols (including many emoji), geometric symbols, arrows, and ornaments originating from the Wingdings and Webdings sets
- Twenty-three new lesser-used and historic scripts extending support for written languages of North America, China, India, other Asian countries, and Africa
- Letters used in Teuthonista and other transcriptional systems, and a new notational set, Duployan

Omitting % and @ on Hash and Array Names

- You used to be able to omit sigils in some circumstances
- Very stupid thing to do
- Deprecation warning since 5.000 (1994!)
- Now a fatal error

`defined(@array)` and `defined(%hash)` now Fatal Errors

- Another parser bug fixed
- Deprecated since 5.6.1
- Deprecation warning since 5.16
- Now a fatal error

`%hash->{ $key } and @array->[$idx]` now Fatal Errors

- Bet you didn't know you could do that
- Now you can't
- Deprecated since "before 5.8"
- Now a fatal error

perlunicook

- New cookbook of Unicode recipes
- By Tom Christiansen
- Well worth reading
- If you think you don't care about Unicode,
you're wrong
- Unicode complicates everything

find2perl, a2p, s2p Removed

- Old programs that converted from standard Unix utilities
- No-one had used them for years
- No-one had maintained them for years
- No-one cares they're gone

CGI Removed

- Contentious
- CGI is a terrible way to write web applications
- Including CGI.pm with Perl was seen as an endorsement
- Best to remove it
- Use PSGI/Plack instead

Module::Build Removed

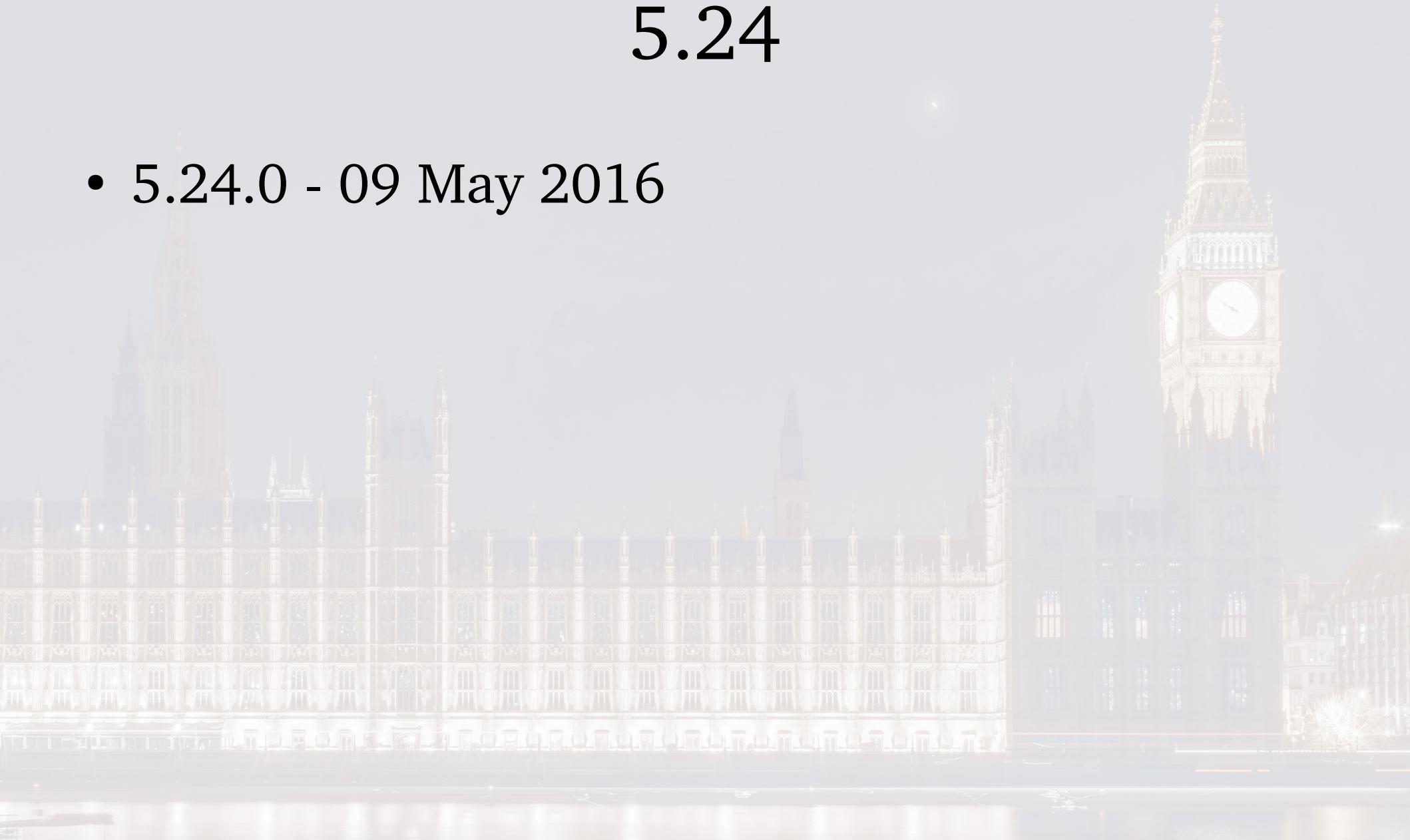
- Written as a replacement for ExtUtils::MakeMaker
- Pure Perl
- Didn't use make
- Cross-platform
- Failed experiment



5.24

5.24

- 5.24.0 - 09 May 2016



5.24 Features

- Postfix dereferencing no longer experimental
- Unicode 8.0
- \b{1b}
- Shebang redirects to Perl6
- autoderef removed

Postfix Dereferencing

- Added in 5.20
- No longer experimental

Unicode 8.0

- Unicode complicates everything!
- <http://www.unicode.org/versions/Unicode8.0.0>
- "Unicode 8.0 adds a total of 7,716 characters, encompassing six new scripts and many new symbols, as well as character additions to several existing scripts."

Unicode 8.0

- A set of lowercase Cherokee syllables, forming case pairs with the existing Cherokee characters
- A large collection of CJK unified ideographs
- Emoji symbols and symbol modifiers for implementing skin tone diversity
- Georgian lari currency symbol
- Letters to support the Ik language in Uganda, Kulango in the Côte d'Ivoire, and other languages of Africa
- The Ahom script for support of the Tai Ahom language in India
- Arabic letters to support Arwi—the Tamil language written in the Arabic script

`\b{lb}`

- New regex boundary
- Matches a (potential) lib break
- Place where you could break a line
- Unicode property
- See also `Unicode::Linebreak`

Shebang Redirects to Perl 6

- Perl will start another program given in the shebang
- E.g. #!/usr/bin/ruby
- Only if the path doesn't contain "perl"
- Now works for "perl6" too

Auto-deref Removed

- Added in 5.14
- Array and hash functions work on references
- `push @$arr_ref` vs `push $arr_ref`
- `keys %$hash_ref` vs `keys %$hash_ref`
- "Deemed unsuccessful"



5.26

Perl 5.26

- Released next spring
- Perl 5.25 is dev branch
- A sneak peek

Perl 5.26

- Lexical subroutines no longer experimental
- Unicode 9.0.0
- Declare references to variables - my \ \$x

Staying Up To Date



Staying Up To Date

- How do you know a new version of Perl has been released?
- How do you know what is in a new release of Perl?

Keep Up To Date With Perl News

- @perl.foundation
- Reddit
- Perl Weekly

@perlfoundation

- The Perl Foundation Twitter account
- ~7,000 followers
- Run by volunteers
- Tries to be first with Perl news

Reddit

- Perl subreddit
- <http://reddit.com/r/perl>
- Links to interesting Perl articles/announcements
- Also help students with their Perl homework

Perl Weekly

- Weekly Perl email
- Links to interesting Perl articles/announcements
- Curated by humans
- Web feed
- Highly recommended

What Is In The New Release?

- Read the perldelta man page
- Included with every Perl release
- Older versions available too

Conclusion

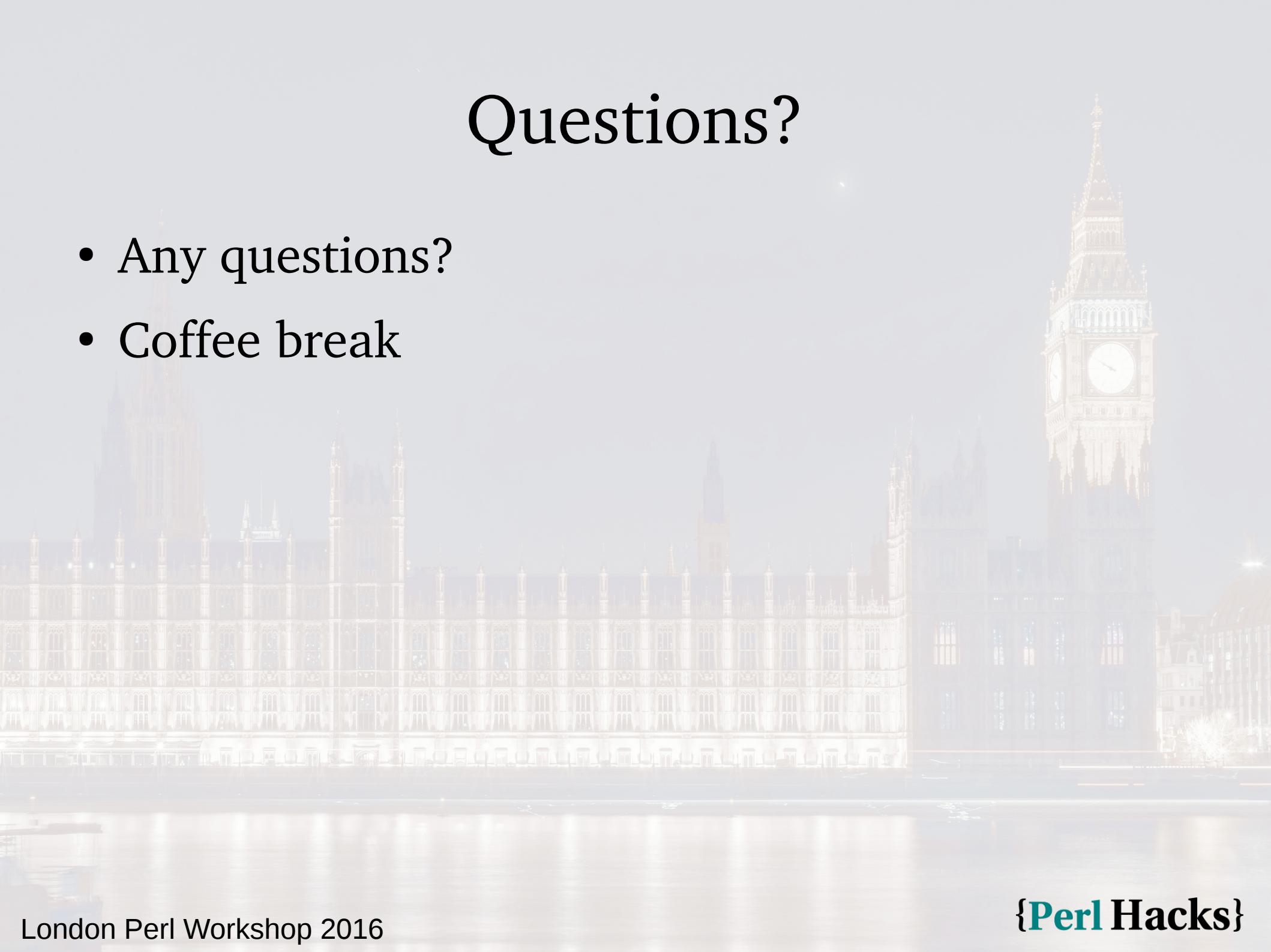


Conclusion

- Perl has an annual release cycle
- Interesting/useful changes in each release
- Worth keeping up to date
- A reason to upgrade?

Questions?

- Any questions?
- Coffee break



Advert

- Magnum Solutions Ltd
- Perl Training
- Perl Development
- Code Review
- Design Review
- dave@mag-sol.com



London Perl Workshop 2016

{Perl Hacks}