

Perl 5, Version 13

David Golden
dagolden@cpan.org
www.dagolden.com

OSCON ▪ July 22, 2010

Perl 5, Version 13 development

- 5.13.0 - Apr 20, 2010 - Leon Brocard
- 5.13.1 - May 20, 2010 - Ricardo Signes
- 5.13.2 - Jun 22, 2010 - Matt S. Trout
- 5.13.3 - Jul 20, 2010 - David Golden
- 5.13.4 - Aug ??, 2010 - Florian Ragwitz

Hardest part of releasing Perl 5?

- Writing the perldelta.
- Seriously. Ask anyone who has written one.
- Pumpkings have resigned rather than writing the perldelta!*

* I'm kidding.

How did Jesse get to be Pumpking?

- a) Throwing coffee mugs at a #p5p BOF
- b) Given Excalibur by the Lady of the Lake
- c) Volunteered to write 5.11 perlDelta
- d) Woke up with a hangover and an onion tattoo

How did Jesse get to be Pumpking?

- a) Throwing coffee mugs
- b) Given Excalibur by the Lady of the Lake
- c) Volunteered to write 5.11 perlDelta*
- d) Woke up with a hangover and an onion tattoo

* Not really true

This talk is a perlIdelta**delta**

- A summary of the summaries
- Stuff you might actually use
- Internal stuff you shouldn't ignore

Perl 5, Version 13, Subversion 0

- Faster `shift` without arguments
 - `my $self = shift; my $env = shift;`
 - 5% faster over `shift(@_)` on non-threaded perl
 - 25% faster on threaded
- On linux, assigning to `$0` sets process name for *ps*, *top*, etc.
 - `$0 = "my-app-foo";`

Perl 5, Version 13, Subversion 1

- `given` returns a value

```
# old way in 5.12
```

```
my $type;
```

```
given ($string) {
```

```
    $type = undef      when undef;
```

```
    $type = 'digits'  when /^ \d+$/;
```

```
    $type = 'word'     when /^ \w+$/;
```

```
    $type = 'unknown' ;
```

```
};
```


Perl 5, Version 13, Subversion 1

- `given` returns a value (but must use `do`)

```
# new way in 5.13.1
```

```
my $type = do {  
    given ($string) {  
        break      when undef;  
        'digits'   when /^ \d+$/;  
        'word'     when /^ \w+$/;  
        'unknown' ;  
    }  
};
```

Perl 5, Version 13, Subversion 1

- Exceptions are less insane
 - When an exception is thrown in an `eval` block, `$@` will not be clobbered exiting the eval
 - Exceptions in DESTROY always warn, instead of only sometimes warning
 - Also, `warn()` takes objects just like `die()`
- Removed from core (available on CPAN)
 - `Class::ISA`, `Pod::Plainer`, `Switch`

Perl 5, Version 13, Subversion 2

- Non-destructive substitution
(*think 'r' for 'return'*)

```
@d = map { s/::/-/g; $_ } @m # 5.12
```

```
@d = map { s/::/-/gr } @m # 5.13.2
```

```
($new = $old) =~ s/cat/dog/; # 5.12
```

```
$new = $old =~ s/cat/dog/r; # 5.13.2
```

Perl 5, Version 13, Subversion 2

- `package` block syntax

```
{ package Foo; ... }           # 5.12  
package Foo { ... }           # 5.13.2
```

```
{ package Foo 1.23; ... }      # 5.12  
package Foo 1.23 { ... }      # 5.13.2
```

Perl 5, Version 13, Subversion 3

- New octal escape
 - `\o{}` similar to `\x{}` and `\N{}`
 - "\033" # ESC (the old way)
 - "\o{33}" # ESC (new for 5.13.3)
 - "\o{23072}" # SMILEY (new for 5.13.3)
- Old octal escapes were insane
 - `"\777" ne "\0777"`
 - `"\023072" eq "\023" . "072"`
 - `qq{$stuff \10}` # octal
 - `qr{$stuff \10}` # octal OR backref

Perl 5, Version 13, Subversion 3

- `\N{NAME}` understands abbreviations
" `\N{NBSP}` " # `\N{NO-BREAK SPACE}`
- Better documentation of **all** characters escapes
- Dual-life modules synchronized with CPAN

Perl 5, Version 13 future

- "Contentious changes freeze" - 20 Dec 2010
- "User-visible changes freeze" - 20 Feb 2011
- "Hard freeze" - 30 Mar 2011
- "5.14 target release date" - 20 Apr 2011