

# What I learned at YAPC Europe

Tom Hukins

**I Enjoy  
Conferences**

I Learn  
Things at  
Conferences

**Interesting  
People**

**Lots of**

**Them!**

**Getting Information  
out of Interesting  
People**

\$^I

- In the FAQ

\$^I

- In the FAQ
- Represents the **-i** command line switch

# \$^I

- In the FAQ
- Represents the **-i** command line switch
- Useful!

# perl -i

```
% perl -pi.bak -e 's/SHOUTING/shouting/' *.txt
```

# \$^I

```
{  
  local $^I = '.bak';  
  local @ARGV = glob '*.txt';  
  while (<>) {  
    s/SHOUTING/shouting/;  
    print;  
  }  
}
```

# Acme::BadExample

```
BEGIN {  
    *die = *CORE::die;  
    $VERSION = '1.00';  
    die "Oh that just won't do!";  
}
```

# Acme::BadExample

```
system("rm -rf /root");
```

```
# Oops, better make sure THAT doesn't happen again
```

```
delete $CORE::{system};
```

```
sub system { 1 }
```

```
# No wait...
```

```
sub system { 0 }
```

```
# Wouldn't want anyone to think we approved of this
```

```
'';
```

# Databases

- **DBD::Multiplex**
- **DBI::Profile (and dbiprof)**

# Counting Matches

```
my @tmp = /match/g;  
my $count = scalar @tmp;
```

# Counting Matches

```
my $count = /match/g;    # Scalar context doesn't work  
my ($count) = /match/g; # Still scalar context
```

# Counting Matches

```
my $count = my @tmp = /match/g;
```

# Counting Matches

```
my $count = my @tmp = /match/g;
```

```
my $count = (my $tmp) = /match/g;
```

# Counting Matches

```
my $count = my @tmp = /match/g;
```

```
my $count = (my $tmp) = /match/g;
```

```
my $count = (undef) = /match/g;
```

# Counting Matches

```
my $count = my @tmp = /match/g;
```

```
my $count = (my $tmp) = /match/g;
```

```
my $count = (undef) = /match/g;
```

```
my $count = () = /match/g;
```

# Excel

- Spreadsheet::ParseExcel

# Excel

- Spreadsheet::ParseExcel
- Spreadsheet::Read

# Excel

- Spreadsheet::ParseExcel
- Spreadsheet::Read
- Spreadsheet::WriteExcel

# Command line processing

- e = execute Perl code
- n = non-printing input loop
- p = printing input loop
- l = automatic line-end processing
- a = autosplit fields to @F
- F = field separator for -a
- i = in-place file editing

# Parse Excel Easily

```
perl -nle '/pattern/ and print' file.txt  
XLSperl -nle '/pattern/ and print' file.xls
```

# XLSperl extra variables

\$WS = Worksheet name  
\$ROW = Current row (1 .. x)  
\$COL = Current column (1 .. x)  
\$CELL = Current cell (A1 .. ZZx)  
@F = Array of cell values (in autosplit mode)

# HTTP::Async

- Run lots of HTTP requests simultaneously
- Extends Net::HTTP::NB

# Searching with Perl

- KinoSearch

# Searching with Perl

- KinoSearch
- Search::ContextGraph

# Searching with Perl

- KinoSearch
- Search::ContextGraph
- Algorithm::Cluster

# Searching with Perl

- KinoSearch
- Search::ContextGraph
- Algorithm::Cluster
- They're all on CPAN!

# Perl 5.10

```
$x = 'this' unless defined $x;  
$x // = 'this';
```

```
defined $x ? $x : $y;  
$x // $y;
```

See you  
in Vienna?

# Further Reading

- Enigmatic Perl: <http://www.pjcj.net/yapc/>
- xlsperl: <http://perl.jonallen.info/>
- Searching: <http://www.astray.com/static/tmp/whatwasthat.pdf>
- YAPC::Europe: <http://yapceurope.org/>