

Perl 5.10

CPAN

perl-5.10.0-RC1

```
#!/per15.10/bin/perl
```

```
> per15.10 script.pl
```

use feature

```
use feature qw(  
    say  
    switch  
    state  
);
```

```
use feature qw(  
    say  
    switch  
    state  
);
```

```
use feature "5.10";
```

```
use feature qw(  
    say  
    switch  
    state  
);
```

```
use feature "5.10";  
use 5.10.0;
```

```
use feature qw(  
    say  
    switch  
    state  
);
```

```
use feature "5.10";  
use v5.10.0;
```

```
#!/per15.10/bin/perl
```

```
use feature "***";
```

```
#!/per15.10/bin/perl
```

```
use feature "say";
```

```
say "Perl 6?";
```

```
#!/per15.10/bin/perl
```

```
use feature "say";
```

```
say "Perl 6?";
```

```
no feature "say";
```

```
> perl5.10 -e \  
"use feature qw(say); say $$;"
```

```
> perl5.10 -e \  
"use feature qw(say); say $$;"
```

```
> perl5.10 -E "say $$;"
```

//

defined-or

```
my $a;
```

```
my $b = $a // 2;
```

```
say $b; 2
```

```
my $c = 0;
```

```
my $d = $c // 3;
```

```
say $d; 0
```

```
my $e = 0;
```

```
my $f = $e || 4;
```

```
say $f; 4
```

my \$_;

```
for (1..5) {  
    my $_ = '*';  
    print;  
}
```

\$:: : _

```
for (1..5) {  
    my $_ = '*';  
    print $::$_;  
}
```

12345

our \$ _ ;

```
for (1..5) {  
  our $_ = '*';  
  print $::_;  
}
```

```
> perl5.10 -E \  
"say for 1..3"
```

```
> perl6 -e \  
"say for 1..3"
```

> perl5.10 -E \ 1

"say for 1..3" 2

3

> perl6 -e \ \n

"say for 1..3" \n

\n

```
use strict 'refs';
```

```
my $x = '***';
```

```
print 1 if defined $$x;
```

```
use strict 'refs';  
my $var = '***';  
print defined $$var  
      ? 'yes' : 'no';
```

```
> perl5.8.8 test.pl
```

no

```
use strict 'refs';  
my $var = '***';  
print defined $$var  
      ? 'yes' : 'no';
```

```
> perl5.10 test.pl
```

Can't use string ("xxx") as a
SCALAR ref while "strict refs"
in use at test.pl line 3.



```
use feature 'say';
```

```
my $date
```

```
    = 'Thu 22 November, 2007';
```

```
say 'Today' if $date =~ /Thu/;
```

```
use feature 'switch';
```

```
use feature qw(switch say);
```

```
my $tag = 'perlrus08';
```

```
given ($tag) {
```

```
    when ('perlrus08') {
```

```
        say 'Yes';
```

```
    }
```

```
}
```

```
use feature qw(switch say);
```

```
my $tag = 'perlrus08';
```

```
given ($tag) {
```

```
    when ('perlrus08') {
```

```
        say 'Yes';
```

```
    }
```

```
    default {say 'No';}
```

```
}
```

when (123)

when (\$value)

when (undef)

when ([2001..2100])

when (/d+/)

when ($\$_{}$ > 0)

when (int)

when (int $\$_{}$)

when (&test_the_value)

when (test_the_value($\$_{}$))

```
given ('perlrus08') {  
    when (/^d+$/) {  
        say 'digits';  
        continue;  
    }  
    when (/perl/i) {  
        say 'Perl';  
    }  
}
```

when (\$what)

==

when (\$_ == \$what)

$\$left \sim\sim \$right$

$= =$

$\$right \sim\sim \$left$

```
use feature 'state';
```

```
sub counter{  
  state $value = 0;  
  $value++;  
  say $value;  
}
```

```
counter();      1  
counter();      2  
counter();      3
```

Регулярные выражения

Именованные буферы

```
my $date = 'Thu 22 November 2007';
```

```
$date =~ /
```

```
(      \w+      )      \s+
```

```
(      \d+      )      \s+
```

```
(      \w+      )      \s+
```

```
(      \d{4}     )
```

```
/x;
```

```
say $1;
```

Thu

```
say $4;
```

2007

Именованные буферы

```
my $date = 'Thu 22 November 2007';
```

```
$date =~ /
```

```
    (?<wday>    \w+    )    \s+
```

```
    (?<day>     \d+    )    \s+
```

```
    (?<month>   \w+    )    \s+
```

```
    (?<year>    \d{4}  )
```

```
/x;
```

```
say $+{wday};
```

Thu

```
say $+{year};
```

2007

Именованные буферы

```
my $date = 'Thu 22 November 2007';
```

```
$date =~ s/  
    (?<year>\d{4})  
    /  
    ${year} + 1  
    /xe;
```

```
say $date;           Thu 22 November 2008
```

```
my $code = 'my $value = 100; say $value;';
```

```
$code =~ s/
```

```
my \s*  
(?<variable> \#[a-z]+) \s*  
= \s*  
(?<value> [^;]+ ) \s*  
; \s*
```

```
(?<other_code>.*?)
```

```
(\k<variable>)
```

```
/$+{other_code}$+{value}/x;
```

```
say $code;
```

```
say 100;
```

```
my $leap_years = '1992 1996 2004 2008';
```

```
$leap_years =~ m/
```

```
(
```

```
    ?<year>    1    \d{3}
```

```
)
```

```
\s*
```

```
(
```

```
    ?<year>    2    \d{3}
```

```
)
```

```
/x;
```

```
say $_ for @{$-}{year};
```

1996

2004

```
my $leap_years = '1992 1996 2004 2008';
```

```
$leap_years =~ m/
```

```
(
```

```
    ?<year>      1      \d{3}
```

```
)
```

```
/gx;
```

```
say $_ for @{$leap_years};
```

```
1992
```

```
my $leap_years = '1992 1996 2004 2008';
```

```
$leap_years =~ m/
```

```
(
```

```
    ?<year>      1      \d{3}      \s*
```

```
)+
```

```
/gx;
```

```
say $_ for @{$-{year}};
```

```
1996
```

```
use feature 'say';
```

```
my $expr = '1 + (2 + (3 + (4 + 5) + 6))';
```

```
$expr =~ s/
```

```
  \ (
```

```
    (
```

```
      [^()]+
```

```
    )
```

```
  |
```

```
    (?1)
```

```
  \)
```

```
  /say $1;/xge;
```

Possessive quantifiers

?+

*+

++

{min, max}+

/

"

(?:

[^"\\]+

|

\\.

)*

"

/x

(? | . . .)

```
my $re = qr/  
    (\d{4})(\d\d)(\d\d)  
    |  
    (\w+),\s*(\d{4})  
/x;
```

```
'20071122' =~ $re;  
say "$1 . $2 . $3";
```

```
'November, 2007' =~ $re;  
say "$4 . $5";
```

```
my $re = qr/  
  (? | (\d{4})(\d\d)(\d\d)  
  |  
    (\w+), \s*(\d{4}))  
/x;
```

```
'20071122' =~ $re;  
say "$1 . $2 . $3";
```

```
'November, 2007' =~ $re;  
say "$1 . $2";
```

$\backslash g\{N\}$

$\backslash gN$

$\setminus g\{-N\}$

`\k<named>`

`==`

`\g{named}`

\K

\v

\h

\V

\H

\R

\R

(?>

\x0D\x0A?

|

[

\x0A-\x0C

\x85

\x{2028}

\x{2029}

]

)

__END__

__DATA__

Андрей Шитов – 2007

andy@shitov.ru | <http://shitov.ru>