Фичи Perl 5.10 на практике

5.10 RC1	17 ноября 2007
5.8.8	1 февраля 2006
5.8.6	27 ноября 2004
5.8.0	18 июля 2002
5.6.0	22 марта 2000

```
#!/usr/bin/perl5.10
```

```
> perl5.10 script.pl
```

```
#!/perl5.10/bin/perl
```

```
> perl5.10 script.pl
```

```
#!/usr/bin/perl
```

```
> perl 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;
```

```
#!/usr/bin/perl

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

```
#!/usr/bin/perl

use feature "say";
say "Perl 6?";
```

```
#!/usr/bin/perl

use feature "say";
say "Perl 6?";
no feature "say";
```

```
> perl -e \
  "use feature qw(say); say $$;"
```

```
> perl -e \
  "use feature qw(say); say $$;"

> perl -E "say $$;"
```

```
> perl -e \
  "use feature qw(say); say $$;"

> perl -E "say $$;"
```

Modern::Perl

```
#!/usr/bin/perl

use Modern::Perl;

say "Perl Tashkent";
```

```
#!/usr/bin/perl

use Modern::Perl;
use utf8;

say "Perl Tashkent";
```

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
```

```
sub get_tagID {
    my $tag = shift;

    return $tagID;
}
```

```
sub get_tagID {
    my $tag = shift;

    # Попытка найти существующий


    return $tagID;
}
```

```
sub get_tagID {
    my $tag = shift;

    # Попытка найти существующий
    # Если нет, то добавить в базу

    return $tagID;
}
```

```
sub get_tagID {
    my $tag = shift;

    # Попытка найти существующий
    # Если нет, то добавить в базу
    # Узнать последний id


    return $tagID;
}
```

```
# Попытка найти существующий
my $sth = dbh->prepare("
    select id from tags
    where tag = ?
")
$sth->execute($tag);

my ($tagID) =
    $sth->fetchrow_array();
$sth->finish();
```

```
# Если нет, то добавить в базу
unless ($tag_ID) {
     my $sth = dbh->prepare("
          insert into tags
          (tag) values (?)
     ");
     $sth->execute($tag);

     . . .
}
```

```
# Узнать последний id
unless ($tag_ID) {
      . . .
     $sth = dbh->prepare("
          select last_insert_id()
          from tags
     ");
     ($tagID) =
          $sth->fetchrow_array();
     $sth->finish();
}
```

```
# Узнать последний id
unless ($tag_ID) {
      . . .
     $sth = dbh->prepare("
          select last_insert_id()
          from tags
     ");
     ($tagID) =
          $sth->fetchrow_array();
     $sth->finish();
}
```

```
sub get_tagID {
    my $tag = shift;

    . . .
    $tagID //= new_tagID();

    return $tagID;
}
```

```
my $per_page =
    $config{per_page} // 10;
```

```
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 \
  "say for 1..3"
```

```
1
2
3
```

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

```
\n
\n
\n
```

Smart matching

```
use feature 'say';

my $date
     = 'Wed 13 May, 2009';
say 'Today' if $date ~~ /Wed/;
```

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

```
use feature qw(switch say);


my $tag = 'perlrus08';
given ($tag) {
    when ('perlrus08') {
        say 'Yes';
    }
}
```

```
use feature qw(switch say);


my $tag = 'perluz1';
given ($tag) {
     when ('perlz1') {
          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 ~~ $right
```

==

```
$right ~~ $left
```

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

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

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

Регулярные выражения

Именованные буферы

```
my $date = 'Wed 13 May 2009';
$date =~ /
    (           \w+  )    \s+
    (           \d+  )    \s+
    (           \w+  )    \s+
    (           \d{4})
/x;

say $1;                   Thu
say $4;                   2007
```

Именованные буферы

```
my $date = 'Wed 13 May 2009';
$date =~ /
    (?<wday>    \w+  )    \s+
    (?<day>     \d+  )    \s+
    (?<month>   \w+  )    \s+
    (?<year>    \d{4})
/x;

say $+{wday};                Thu
say $+{year};                2007
```

Именованные буферы

```
my $date = 'Wed 13 May 2009';

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

say $date;          Wed 13 May 2009
```

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

Posessive quantifiers

?+

*+

++

{min, max}+

```
/
   "
      (?:
            [^"\\]++
               |
                  \\.
      )*+
   "
/x
```

(?| . . .)

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

'20090513' =~ $re;
say "$1 . $2 . $3";

'May, 2009' =~ $re;
say "$4 . $5";
```

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

'20090513' =~ $re;
say "$1 . $2 . $3";

'May, 2009' =~ $re;
say "$1 . $2";
```

\g{N}

\gN

```
\g{-N}
```

\k<named>

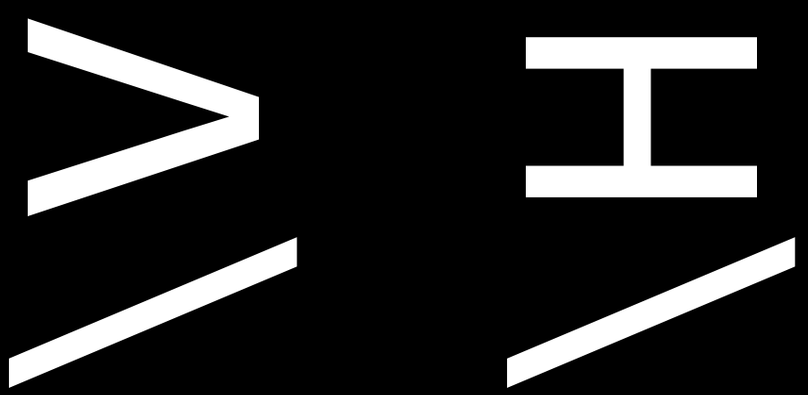
==

\g{named}

\K

\v

\h



\R

```
\R                  (?>
                              \x0D\x0A?
                        |
                              [
                                  \x0A-\x0C
                                  \x85
                                  \x{2028}
                                  \x{2029}
                              ]
                    )
```

END

```
__DATA__
Андрей Шитов — 2007, 2009
andy@shitov.ru  |  http://shitov.ru
```