

[функции](#) / [наш](#) ([Источник](#), [CPAN](#))

список наших вариантов

список наших ТИПОВ

список наших вариантов : ATTRS

список ВАРИАНТОВ нашего ТИПА : ATTRS

`our` создает лексический псевдоним пакетной (т.е. глобальной) переменной с тем же именем в текущем пакете для использования в текущей лексической области видимости.

`our` имеет те же правила определения области, что и `my` или `state`, что означает, что оно допустимо только в лексической области. В отличие от `my` и `state`, которые оба объявляют новые (лексические) переменные, `our` создает псевдоним только для существующей переменной: пакетной переменной с тем же именем.

Это означает, что когда `use strict 'vars'` действует, `our` позволяет использовать переменную пакета, не уточняя ее именем пакета, а только в лексической области `our` объявления. Это применяется немедленно - даже в рамках одного и того же оператора.

```
package Foo;
use v5.36; # which implies "use strict;"

$Foo::foo = 23;

{
    our $foo; # alias to $Foo::foo
    print $foo; # prints 23
}

print $Foo::foo; # prints 23

print $foo; # ERROR: requires explicit package name
```

Это работает, даже если переменная `package` ранее не использовалась, поскольку переменные `package` появляются при первом использовании.

```
package Foo;
use v5.36;

our $foo = 23; # just like $Foo::foo = 23

print $Foo::foo; # prints 23
```

Поскольку переменная немедленно становится легальной в `use strict 'vars'`, пока нет переменной с таким именем, которая уже находится в области видимости, вы можете снова ссылаться на переменную пакета даже в том же операторе.

```
package Foo;
use v5.36;

my $foo = $foo; # error, undeclared $foo on right-hand side
our $foo = $foo; # no errors
```

Если указано более одной переменной, список должен быть заключен в круглые скобки.

```
our($bar, $baz);
```

В объявлении `our` объявляется псевдоним переменной пакета, который будет виден во всей его лексической области видимости, даже за границами пакета. Пакет, в который вводится переменная, определяется в момент объявления, а не в момент использования. Это означает, что выполняется следующее поведение:

```
package Foo;
our $bar; # declares $Foo::bar for rest of lexical scope
$bar = 20;

package Bar;
print $bar; # prints 20, as it refers to $Foo::bar
```

Допускается несколько `our` объявлений с одинаковым именем в одной лексической области видимости, если они находятся в разных пакетах. Если они окажутся в одном пакете, Perl выдаст предупреждения, если вы их запросили, точно так же, как несколько `my` объявлений. В отличие от второго `my` объявления, которое привязывает имя к новой переменной, второе `our` объявление в том же

пакете, в той же области видимости, просто избыточно.

```
use warnings;
package Foo;
our $bar;      # declares $Foo::bar for rest of lexical scope
$bar = 20;

package Bar;
our $bar = 30; # declares $Bar::bar for rest of lexical scope
print $bar;    # prints 30

our $bar;      # emits warning but has no other effect
print $bar;    # still prints 30
```

Объявление our также может содержать список атрибутов, связанных с ним.

Точная семантика и интерфейс TYPE и ATTRS все еще развиваются. В настоящее время тип привязан к использованию прагмы fields, а атрибуты обрабатываются с помощью прагмы attributes или, начиная с Perl 5.8.0, также через модуль Attribute::Handlers. Смотрите "Приватные переменные через my()" в perlsub для получения подробной информации.

Обратите внимание, что список, заключенный в скобки, undef может использоваться как фиктивный заполнитель, например, чтобы пропустить присвоение начальных значений:

```
our ( undef, $min, $hour ) = localtime;
```

our отличается от use vars, который позволяет использовать невалифицированное имя *только* в пределах затронутого пакета, но в разных областях.

Браузер Perldoc поддерживается Дэном Буком ([DBOOK](#)). Пожалуйста, свяжитесь с ним через [GitHub issue tracker](#) или [по электронной почте](#) по поводу любых проблем с самим сайтом, поиском или отображением документации.

Документация по Perl поддерживается разработчиками Perl 5 при разработке Perl. Пожалуйста, свяжитесь с ними через [систему отслеживания проблем с Perl](#), [список рассылки](#) или [IRC](#), чтобы сообщить о любых проблемах с содержанием или форматом документации.