

92.13.7 Arrow operator and arrays

When the left argument to the arrow operator `->` is an array, or the `scalar` operator operating on an array, the result of the operation must be considered erroneous. For example:

```
@x->[2]
scalar(@x)->[2]
```

These expressions will get run-time errors in some future release of Perl.

92.13.8 Experimental features

As discussed above, many features are still experimental. Interfaces and implementation of these features are subject to change, and in extreme cases, even subject to removal in some future release of Perl. These features include the following:

Threads

Unicode

64-bit support

Lvalue subroutines

Weak references

The pseudo-hash data type

The Compiler suite

Internal implementation of file globbing

The DB module

The regular expression code constructs:

```
(?{ code }) and (??{ code })
```

92.14 Obsolete Diagnostics

Character class syntax `[:]` is reserved] for future extensions

(W) Within regular expression character classes (`[]`) the syntax beginning with `["` and ending with `"]` is reserved for future extensions. If you need to represent those character sequences inside a regular expression character class, just quote the square brackets with the backslash: `"\[` and `"]`.

Ill-formed logical name `[%s]` in `prime_env_iter`

(W) A warning peculiar to VMS. A logical name was encountered when preparing to iterate over `%ENV` which violates the syntactic rules governing logical names. Because it cannot be translated normally, it is skipped, and will not appear in `%ENV`. This may be a benign occurrence, as some software packages might directly modify logical name tables and introduce nonstandard names, or it may indicate that a logical name table has been corrupted.

In string, `@%s` now must be written as `\@%s`

The description of this error used to say:

```
(Someday it will simply assume that an unbackslashed @
interpolates an array.)
```