```
# In the main program
push @INC, new Foo(...);
```

Note that these hooks are also permitted to set the %INC entry corresponding to the files they have loaded. See %INC in *perlvar*.

For a yet-more-powerful import facility, see use and *perlmod*.

reset EXPR

reset

Generally used in a `continue` block at the end of a loop to clear variables and reset `??` searches so that they work again. The expression is interpreted as a list of single characters (hyphens allowed for ranges). All variables and arrays beginning with one of those letters are reset to their pristine state. If the expression is omitted, one-match searches (`?pattern?`) are reset to match again. Resets only variables or searches in the current package. Always returns 1. Examples:

```
reset 'X';          # reset all X variables
reset 'a-z';        # reset lower case variables
reset;              # just reset ?one-time? searches
```

Resetting `"A-Z"` is not recommended because you'll wipe out your `@ARGV` and `@INC` arrays and your `%ENV` hash. Resets only package variables–lexical variables are unaffected, but they clean themselves up on scope exit anyway, so you'll probably want to use them instead. See my.

return EXPR

return

Returns from a subroutine, `eval`, or `do FILE` with the value given in EXPR. Evaluation of EXPR may be in list, scalar, or void context, depending on how the return value will be used, and the context may vary from one execution to the next (see `wantarray`). If no EXPR is given, returns an empty list in list context, the undefined value in scalar context, and (of course) nothing at all in a void context.

(Note that in the absence of an explicit `return`, a subroutine, eval, or do FILE will automatically return the value of the last expression evaluated.)

reverse LIST

In list context, returns a list value consisting of the elements of LIST in the opposite order. In scalar context, concatenates the elements of LIST and returns a string value with all characters in the opposite order.

```
print reverse <>;           # line tac, last line first

undef $/;                   # for efficiency of <>
print scalar reverse <>;    # character tac, last line tsrif
```

Used without arguments in scalar context, reverse() reverses $_.

This operator is also handy for inverting a hash, although there are some caveats. If a value is duplicated in the original hash, only one of those can be represented as a key in the inverted hash. Also, this has to unwind one hash and build a whole new one, which may take some time on a large hash, such as from a DBM file.

```
%by_name = reverse %by_address;     # Invert the hash
```

rewinddir DIRHANDLE

Sets the current position to the beginning of the directory for the `readdir` routine on DIRHANDLE.

rindex STR,SUBSTR,POSITION