`ls`
`ls` *`filenames`*
`ls` *`options`*
`ls` *`options filenames`*
`list` `= ls (...)`

List directory contents.

The `ls` command is implemented by calling the native operating system's directory listing command—available *options* will vary from system to system.

Filenames are subject to shell expansion if they contain any wildcard characters '`*`', '`?`', '`[]`'. To find a literal example of a wildcard character the wildcard must be escaped using the backslash operator '\'.

If the optional output *list* is requested then `ls` returns a character array with one row for each file/directory name.

Example usage on a UNIX-like system:

```
ls -l
     ⊣ total 12
     ⊣ -rw-r--r--   1 jwe  users  4488 Aug 19 04:02 foo.m
     ⊣ -rw-r--r--   1 jwe  users  1315 Aug 17 23:14 bar.m
```

See also: [dir], page 805, [readdir], page 786, [glob], page 786, [what], page 134, [stat], page 783, [filesep], page 787, [ls_command], page 805.

`val` `= ls_command ()` [Function File]
`old_val` `= ls_command (`*`new_val`*`)` [Function File]

Query or set the shell command used by Octave's `ls` command.

See also: [ls], page 805.

`dir` [Function File]
`dir (`*`directory`*`)` [Function File]
`[`*`list`*`] = dir (`*`directory`*`)` [Function File]

Display file listing for directory *directory*.

If *directory* is not specified then list the present working directory.

If a return value is requested, return a structure array with the fields

name	File or directory name.
date	Timestamp of file modification (string value).
bytes	File size in bytes.
isdir	True if name is a directory.
datenum	Timestamp of file modification as serial date number (double).
statinfo	Information structure returned from `stat`.

If *directory* is a filename, rather than a directory, then return information about the named file. *directory* may also be a list rather than a single directory or file.

directory is subject to shell expansion if it contains any wildcard characters '`*`', '`?`', '`[]`'. To find a literal example of a wildcard character the wildcard must be escaped using the backslash operator '\'.