



## ls - Listing Files and Directories

### Basic Usage

<code>ls</code>	Lists files and directories in the current directory.
<code>ls &lt;directory&gt;</code>	Lists files and directories in the specified directory.
<code>ls -l</code>	Lists files in long format, showing permissions, size, modification date, etc.
<code>ls -a</code>	Lists all files, including hidden files (those starting with <code>.</code> ).
<code>ls -t</code>	Sorts files by modification time (newest first).
<code>ls -r</code>	Reverses the order of the listing.

### Combining Options

<code>ls -la</code>	Lists all files (including hidden) in long format.
<code>ls -lt</code>	Lists files in long format, sorted by modification time.
<code>ls -lt</code>	Lists files in long format, sorted by modification time (oldest first).
<code>ls -lh</code>	Lists files in long format, with file sizes in human-readable format (e.g., KB, MB, GB).
<code>ls -d</code>	List directories - useful with wildcards.
<code>ls -F</code>	Appends a character to each filename indicating the file type (* for executable, / for directory, @ for symbolic link, = for socket).

### Advanced Usage

<code>ls -R</code>	- Lists subdirectories encountered recursively.
<code>ls -S</code>	- Sort by file size, largest first.
<code>ls -li</code>	- Show the index number of each file.
<code>ls --color[=WHEN]</code>	- Control whether color is used to distinguish file types. <code>WHEN</code> can be <code>always</code> (default if omitted), <code>auto</code> , or <code>never</code> .
<code>ls -b</code>	- Escape non graphic characters as octal numbers.

## cp - Copying Files and Directories

### Basic Copying

<code>cp &lt;source&gt; &lt;destination&gt;</code>	Copies a file or directory from source to destination.
<code>cp file1 file2 file3 &lt;directory&gt;</code>	Copies multiple files to a directory.
<code>cp -r &lt;source_dir&gt; &lt;destination_dir&gt;</code>	Copies a directory recursively (including all files and subdirectories).
<code>cp -i &lt;source&gt; &lt;destination&gt;</code>	Interactive mode. Asks for confirmation before overwriting existing files.
<code>cp -p &lt;source&gt; &lt;destination&gt;</code>	Preserves the original file's mode, ownership, and timestamps.

### Advanced Copying

<code>cp -u &lt;source&gt; &lt;destination&gt;</code>	- Copy only when the SOURCE file is newer than the destination file or when the destination file is missing.
<code>cp -v &lt;source&gt; &lt;destination&gt;</code>	- Verbose mode. Shows files as they are copied.
<code>cp -s &lt;source&gt; &lt;destination&gt;</code>	- Make symbolic links instead of copying.
<code>cp -l &lt;source&gt; &lt;destination&gt;</code>	- Make hard links instead of copying.
<code>cp --backup[=CONTROL] &lt;source&gt; &lt;destination&gt;</code>	- Back up each existing destination file. The CONTROL argument specifies the backup suffix.

### Examples

Copying a file: <code>cp my_file.txt /home/user/documents/</code>
Copying a directory recursively: <code>cp -r my_directory /home/user/backup/</code>
Copying multiple files with confirmation: <code>cp -i file1.txt file2.txt /tmp/</code>

## mv - Moving and Renaming Files and Directories

### Basic Moving and Renaming

<code>mv &lt;source&gt;</code> <code>&lt;destination&gt;</code> <code>&gt;</code>	Moves or renames a file or directory from source to destination.
<code>mv file1</code> <code>file2 file3</code> <code>&lt;directory&gt;</code>	Moves multiple files to a directory.
<code>mv -i</code> <code>&lt;source&gt;</code> <code>&lt;destination&gt;</code> <code>&gt;</code>	Interactive mode. Asks for confirmation before overwriting existing files.

### Moving Options

<code>mv -n</code> <code>&lt;source&gt;</code> <code>&lt;destination&gt;</code>	No clobber. Do not overwrite an existing file. Useful for scripts.
<code>mv -v</code> <code>&lt;source&gt;</code> <code>&lt;destination&gt;</code>	Verbose. Shows files as they are moved.
<code>mv -u</code> <code>&lt;source&gt;</code> <code>&lt;destination&gt;</code>	Move only when the SOURCE file is newer than the destination file or when the destination file is missing.

### Examples

Renaming a file: <code>mv old_file.txt new_file.txt</code>
Moving a file to a different directory: <code>mv my_file.txt /home/user/documents/</code>
Moving multiple files to a directory: <code>mv file1.txt file2.txt /tmp/</code>
Moving a directory: <code>mv my_directory /opt/</code>

## rm - Removing Files and Directories

### Basic Removal

<code>rm</code> <code>&lt;file&gt;</code>	Removes a file.
<code>rm -i</code> <code>&lt;file&gt;</code>	Interactive mode. Asks for confirmation before deleting.
<code>rm -f</code> <code>&lt;file&gt;</code>	Force removal. Ignores nonexistent files and suppresses prompts.

### Removing Directories

<code>rmdir</code> <code>&lt;directory&gt;</code>	Removes an empty directory.
<code>rm -r</code> <code>&lt;directory&gt;</code>	Removes a directory and its contents recursively.
<code>rm -rf</code> <code>&lt;directory&gt;</code>	Force removal of a directory and its contents recursively. Use with caution!

### Examples

Removing a file: <code>rm my_file.txt</code>
Removing a directory recursively: <code>rm -r my_directory</code>
Force removing a directory and its contents: <code>rm -rf my_directory</code>
Removing multiple files: <code>rm file1.txt file2.txt file3.txt</code>

### Important Notes

<b>Caution:</b> <code>rm -rf /</code> will delete everything on the system if run as root. Be extremely careful when using <code>rm</code> with wildcards or the <code>-r</code> and <code>-f</code> options.
<b>Best Practice:</b> Use <code>rm -i</code> to confirm each deletion, especially when using wildcards.