CHMOD QUICK REFERENCE

File permissions, numeric & symbolic modes, special bits, umask

Numeric Mode

Octal Permission Digits

`4`	Read (r)
`2`	Write (w)
`1`	Execute (x)
`0`	No permission

Three-Digit Format

```
chmod 755 file    # rwxr-xr-x
chmod 644 file    # rw-r--r--
chmod 700 file    # rwx------
chmod 600 file    # rw-------
```

Digit Calculation

`7 (4+2+1)`	rwx — read, write, execute
`6 (4+2)`	rw- — read, write
`5 (4+1)`	r-x — read, execute
`4`	r-- — read only
`3 (2+1)`	-wx — write, execute
`2`	-w- — write only
`1`	--x — execute only
`0`	--- — no permissions

Symbolic Mode

Syntax: [ugoa][+-=][rwxXst]

`u`	User (owner)
`g`	Group
`o`	Others
`a`	All (u + g + o)
`+`	Add permission
`-`	Remove permission
`=`	Set exact permission

Symbolic Examples

```
chmod u+x file         # owner: add execute
chmod g-w file         # group: remove write
chmod o=r file         # others: set read only
chmod a+r file         # all: add read
chmod u+x,g-w,o= file # combined operations
```

Common Permissions

File Permission Presets

`644 rw-r--r--`	Default file — owner rw, others read
`755 rwxr-xr-x`	Script / binary — owner rwx, others rx
`600 rw-------`	Private file — owner only
`400 r--------`	Read-only private (SSH keys)
`666 rw-rw-rw-`	World-writable file (avoid)
`777 rwxrwxrwx`	Full access for all (avoid)

Quick Reference

```
chmod 644 *.html       # web files: owner rw, world r
chmod 755 *.sh         # scripts: owner rwx, world rx
chmod 600 ~/.ssh/id_*  # SSH keys: owner only
chmod 400 secret.pem   # certificate: read-only
```

Directory Permissions

What Permissions Mean for Directories

`r (4)`	List directory contents (`ls`)
`w (2)`	Create / delete files in directory
`x (1)`	Access (cd into) the directory
`rx (5)`	List + access (typical for read)
`rwx (7)`	Full control

Common Directory Permissions

```
chmod 755 dir/     # standard: owner rwx, others rx
chmod 700 dir/     # private: owner only
chmod 750 dir/     # group access: owner rwx, group rx
chmod 1777 /tmp    # sticky bit: only owner can delete
```

Special Bits

Setuid, Setgid, Sticky

`Setuid (4xxx)`	Run as file owner (e.g., `passwd`)
`Setgid (2xxx)`	Run as file group / inherit dir group
`Sticky (1xxx)`	Only owner can delete files (e.g., `/tmp`)

Setting Special Bits

```
chmod 4755 program     # setuid: -rwsr-xr-x
chmod 2755 dir/        # setgid: drwxr-sr-x
chmod 1755 dir/        # sticky: drwxr-xr-t
chmod u+s program      # symbolic setuid
chmod g+s dir/         # symbolic setgid
chmod +t dir/          # symbolic sticky bit
```

Recursive

Recursive Permission Changes

```
chmod -R 755 dir/           # set all to 755 recursively
chmod -R u+rwX dir/         # owner rw, +x on dirs only
chmod -R go-w dir/          # remove group/other write
```

Files vs Directories with find

```
# set directories to 755, files to 644
find /path -type d -exec chmod 755 {} +
find /path -type f -exec chmod 644 {} +
```

Capital X — Conditional Execute

`x (lowercase)`	Add execute to all files and dirs
`X (uppercase)`	Add execute only to dirs and already-executable files

umask

How umask Works

`umask`	Display current umask
`umask 022`	Files: 644, Dirs: 755
`umask 077`	Files: 600, Dirs: 700
`umask 002`	Files: 664, Dirs: 775

umask Calculation

```
# default permission minus umask = effective
# Files:  666 - 022 = 644 (rw-r--r--)
# Dirs:   777 - 022 = 755 (rwxr-xr-x)
umask          # display current umask
umask 022      # typical default
umask -S       # show in symbolic notation
```

Common Patterns

Everyday Use Cases

`Web root`	`chmod -R 755 /var/www/html`
`Config file`	`chmod 600 app.conf`
`SSH directory`	`chmod 700 ~/.ssh`
`SSH authorized_keys`	`chmod 600 ~/.ssh/authorized_keys`
`Shared directory`	`chmod 2775 /shared` (setgid)
`Log files`	`chmod 640 /var/log/app.log`
`Cron scripts`	`chmod 755 /etc/cron.daily/myjob`
`Temp directory`	`chmod 1777 /tmp` (sticky)

Viewing Permissions

```
ls -l file.txt       # show permissions
ls -ld dir/          # show directory permissions
stat -c '%A %a %n' * # symbolic + numeric + name
getfacl file.txt     # show ACLs (if in use)
```