🔐 Linux Permissions Cheat Sheet

📊 Permission Types

Symbol	Permission	Number	Meaning
r	Read	4	View file contents
w	Write	2	Modify file
x	Execute	1	Run as program
-	None	0	No permission

👥 Permission Groups

-rwxrwxr-x

```
|||||||||
||||||||└─ Others: Execute
|||||||└── Others: Write
||||||└─── Others: Read
|||||└──── Group: Execute
||||└───── Group: Write
|||└────── Group: Read
||└─────── Owner: Execute
|└──────── Owner: Write
└───────── Owner: Read
```

🔢 Numeric Permissions (Quick Reference)

Number	Permission	Symbolic
0	No permission	---
1	Execute only	--x
2	Write only	-w-
3	Write + Execute	-wx

Number	Permission	Symbolic
4	Read only	r--
5	Read + Execute	r-x
6	Read + Write	rw-
7	All permissions	rwx

Calculate: r(4) + w(2) + x(1) = permission number

⚡ **Common chmod Commands**

Symbolic Mode (Easy to Remember)

```
# Add permissions
chmod +x file.sh        # Add execute for everyone
chmod +r file.txt       # Add read for everyone
chmod +w file.txt       # Add write for everyone

# Remove permissions
chmod -x file.sh        # Remove execute from everyone
chmod -w file.txt       # Remove write from everyone

# Specific groups
chmod u+x file.sh       # Add execute for User (owner)
chmod g+w file.txt      # Add write for Group
chmod o-r file.txt      # Remove read from Others
chmod a+x file.sh       # Add execute for All

# Set exact permissions
chmod u=rwx file.sh     # User: read+write+execute
chmod g=rx file.sh      # Group: read+execute
chmod o=r file.sh       # Others: read only
```

```
# Multiple changes at once
chmod u+x,g+x,o+r file    # Add execute for user & group, read for others
chmod u=rwx,g=rx,o=r file # Set exact permissions for all groups
```

🎯 Most Common Permission Sets

Command	Numeric	Result	Use Case
chmod +x file	chmod 755 file	rwxr-xr-x	**Scripts** - Owner can edit, everyone can run
chmod 644 file	chmod 644 file	rw-r--r--	**Regular files** - Owner can edit, others can read
chmod 755 dir	chmod 755 dir	rwxr-xr-x	**Directories** - Owner can modify, others can access
chmod 700 file	chmod 700 file	rwx------	**Private scripts** - Only owner can access
chmod 600 file	chmod 600 file	rw-------	**Private files** (SSH keys, passwords)
chmod 777 file	chmod 777 file	rwxrwxrwx	⚠️ **DANGEROUS** - Everyone can do everything
chmod 444 file	chmod 444 file	r--r--r--	**Read-only** - No one can modify

---

🚀 Quick Commands

```
# View permissions
ls -l file.txt              # Long format with permissions
stat file.txt               # Detailed permission info

# Change ownership
chown user:group file.txt     # Change owner and group
chown user file.txt           # Change owner only
chgrp group file.txt          # Change group only

# Recursive changes
chmod -R 755 directory/       # Apply to directory and all contents
```

```
chown -R user:group directory/    # Change ownership recursively


# Check current user
whoami                  # Your username
groups                  # Your groups
id                      # Detailed user info
```

📁 Directory Permissions

Directories need different permissions than files!

Permission	On Files	On Directories
r	Read contents	List files inside (ls)
w	Modify file	Create/delete files inside
x	Execute file	Enter directory (cd)

Example:

```
chmod 755 mydir/          # Standard directory permissions
# rwxr-xr-x
# Owner: can enter, list, and modify contents
# Others: can enter and list, but not modify
```

🎨 Terminal Color Codes

Color	File Type
White/Black	Regular file (not executable)
Green	Executable file
Blue	Directory
Cyan	Symbolic link
Red	Archive file (.zip, .tar.gz)
Yellow	Device file

Color	File Type
Magenta	Image/media file

⚡ One-Liner Fixes

```
# Fix common script issues
chmod +x script.sh && ./script.sh

# Make all .sh files executable
chmod +x *.sh

# Reset file to standard permissions
chmod 644 file.txt

# Reset script to standard permissions
chmod 755 script.sh

# Make SSH key secure (required by SSH)
chmod 600 ~/.ssh/id_rsa

# Fix "Permission denied" when running script
chmod u+x script.sh
```

🔍 Permission Checking

```
# Check if file is executable
if [ -x script.sh ]; then
    echo "Executable"
fi
```

```
# Check if file is readable
if [ -r file.txt ]; then
   echo "Readable"
fi

# Check if file is writable
if [ -w file.txt ]; then
   echo "Writable"
fi
```

🛡️ Security Best Practices

✅ Scripts: `chmod 755 script.sh` # rwxr-xr-x

✅ Config files: `chmod 644 config.conf` # rw-r--r--

✅ SSH keys: `chmod 600 ~/.ssh/id_rsa` # rw-------

✅ Directories: `chmod 755 mydir/` # rwxr-xr-x

✅ Private scripts: `chmod 700 private.sh` # rwx------

❌ NEVER use: `chmod 777 anything` # Too permissive!

📝 Quick Examples

```
# Create executable script
nano script.sh
chmod +x script.sh
./script.sh

# Fix permissions on all scripts in directory
chmod +x *.sh
```

Make file read-only for everyone

chmod 444 important.txt

Secure private directory

chmod 700 private_folder/

Share directory (read-only for others)

chmod 755 shared_folder/

Remove all permissions for others

chmod o-rwx secret_file

🎯 **Remember This Pattern**

chmod [who][action][permission] filename

who: u (user/owner) g (group) o (others) a (all)

action: + (add) - (remove) = (set exactly)

permission: r (read) w (write) x (execute)

💡 **Mnemonics to Remember**

- **755** = "Scripts Standard" - **S**cripts **S**hould have **755**
- **644** = "Files Forever Friendly" - Regular **F**iles **F**or **644**
- **700** = "Private is Perfect" - **P**rivate **P**ermissions = **700**
- **777** = "Terrible Terrible Terrible" - **Never** use!

Quick Test Your Understanding:

-rwxr-xr-x = 755 = Owner: all, Others: read+execute

-rw-r--r-- = 644 = Owner: read+write, Others: read only

-rwx------ = 700 = Owner: all, Others: nothing