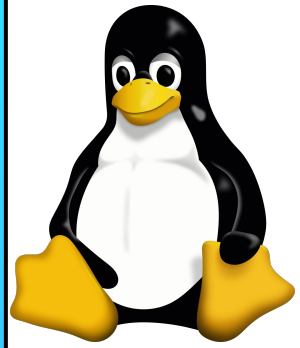


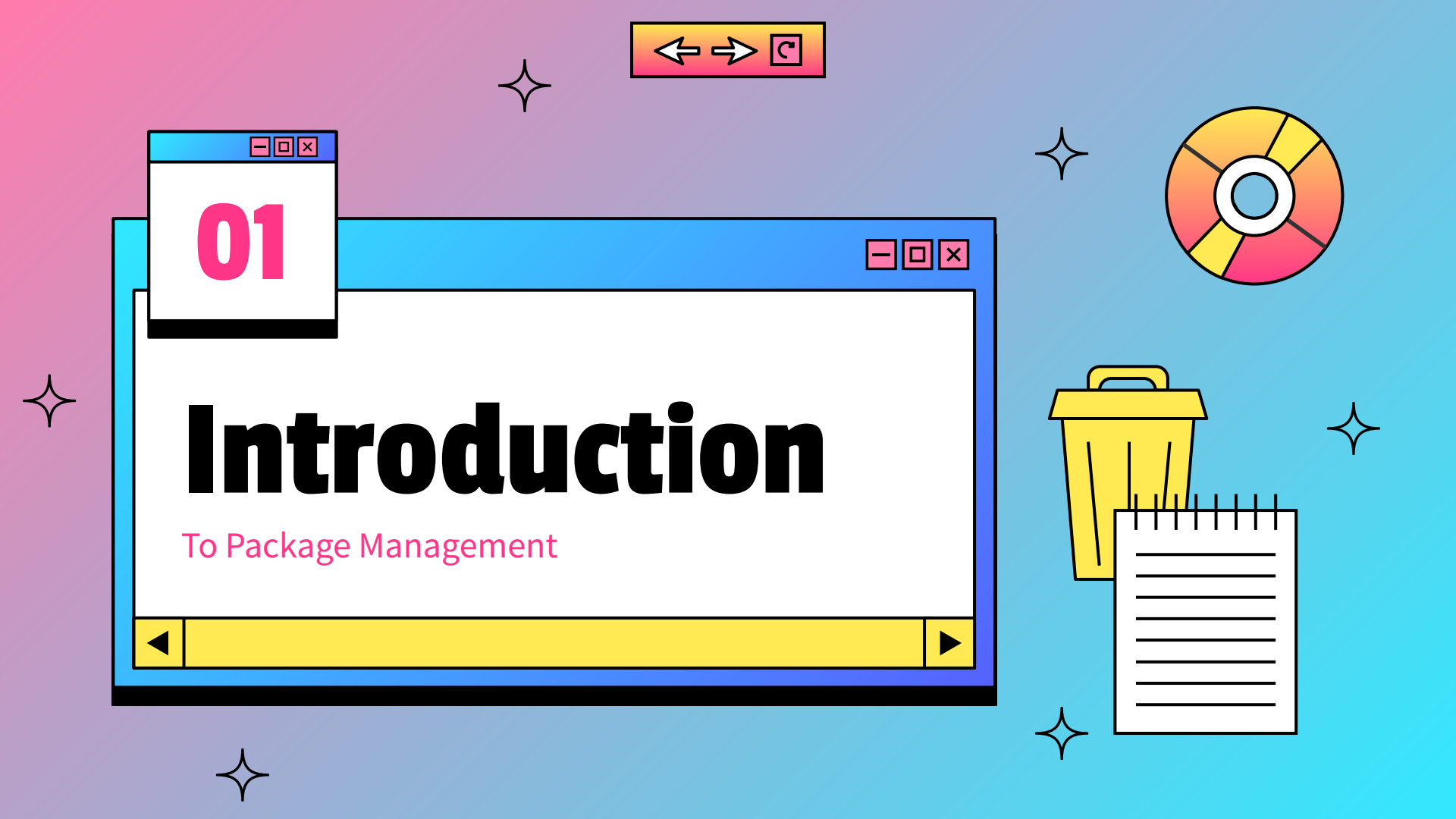
The Linux Week

Introduction to Linux

Welcome to Day 5

Day 5

- Introduction to Package Management
  - Introduction
  - Key Concepts
- Popular Package Managers
  - APT
  - DNF
  - YUM
  - Zypper
  - Pacman
- APT (Advanced Package Tool)
- Real World Applications


01
Introduction
To Package Management

Provide a standardized way to handle software packages, ensuring that software can be easily installed, updated, and removed

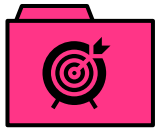
Package Management

Key Concepts

Repositories

Central locations where packages are stored and maintained.

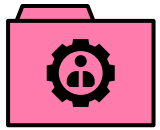
1



Dependencies

Packages required for a particular package to function correctly

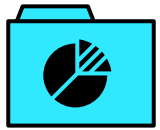
3



2

Packages

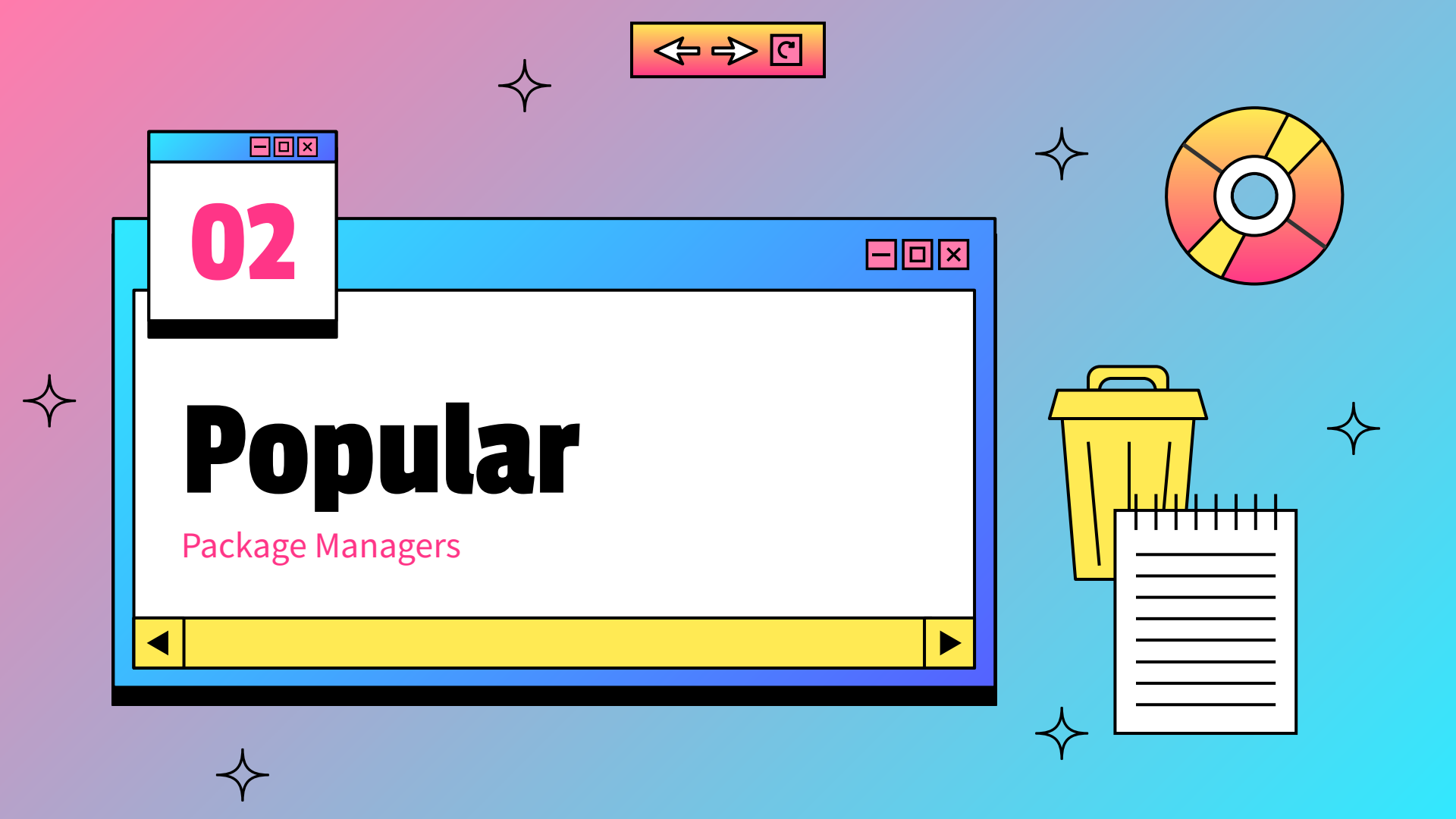
Bundled software applications or libraries, typically in compressed archive formats,



4

Meta Data

Information about packages, including version numbers, and descriptions


02
Popular
Package Managers

APT

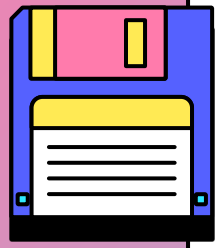
Advanced Package Tool

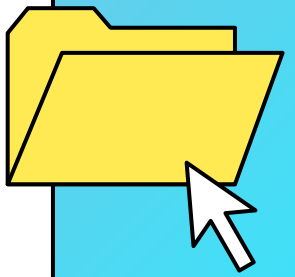
Used by Debian and Debian based distributions like Ubuntu

DNF

Dandified YUM

Used by Fedora, Red Hat Enterprise Linux (RHEL) and CentOS





YUM

Yellowdog Updater, Modified

Older versions of Fedora, RHEL, CentOS

ZYPPER

Zypper

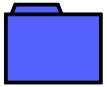
Used by OpenSUSE and SUSE Distributions

Pacman

Pacman

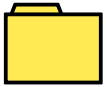
Used by Arch Linux and Arch Linux Distributions

Functions of Package Managers



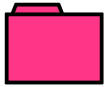
Installation

Installing software by resolving dependencies and fetching packages from repositories



Upgrades

Keeping users' systems up-to-date by upgrading packages to their latest versions.



Removal

Safely removes packages and their associated files

Functions of Package Managers

Dependency Resolution

Automatically identifies and installs any dependencies required by a package.

Repositories Management

Adds or removes repositories to expand the range of available software.

03

APT - Debian

Package Manager

Key Commands

Apt-get update

Updates the package list

Apt-get upgrade

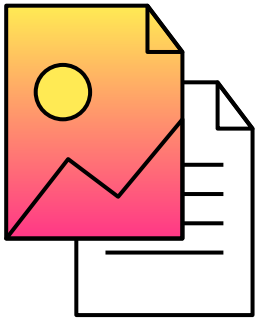
Upgrades all installed packages

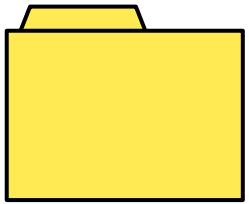
apt-get install <package>

Installs a specific package

apt-get remove <package>

Removes a specific package





Figlet Package

1- apt-cache search figlet
2- sudo apt-get install figlet
3- figlet Awesome!

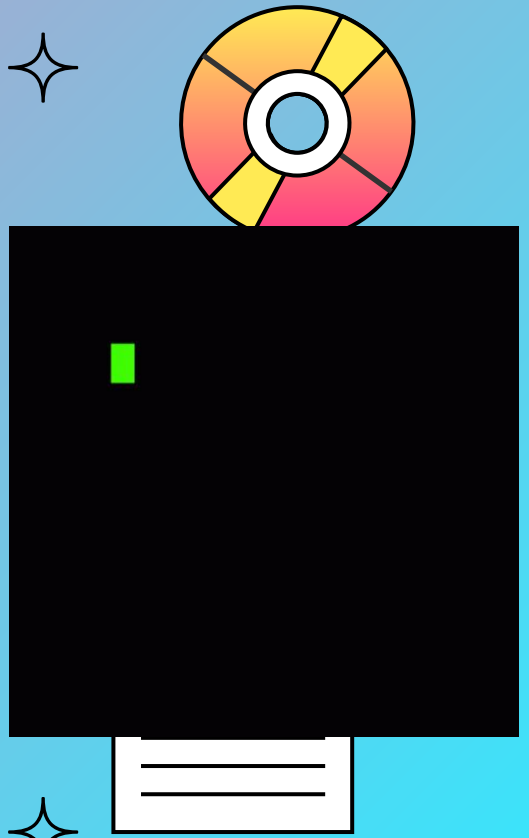
Cowsay Package

1- apt-cache search speaking cow
2- sudo apt-get install cowsay
3- cowsay “hello!”

04

Real-World

Applications and Use cases



Connecting Apache Web Server with MySQL Database Using Ubuntu Terminal

1) Installations of Packages

Commands:

sudo apt-get update

sudo apt-get install apache2 mysql-server php libapache2-mod-php php-mysql

Explanation:
sudo apt-get update: Updates the package list.

sudo apt-get install apache2 mysql-server php libapache2-mod-php php-mysql: Installs Apache, MySQL, PHP, and PHP-MySQL extension.

2) Secure MySQL Installation

Command:

sudo mysql_secure_installation

Explanation:

This command secures MySQL by setting a root password and removing insecure defaults.

3) Create a MySQL Database and User

Command:

sudo mysql -u root -p

Explanation:

Enter your MySQL root password when prompted.

3) Create a MySQL Database and User

In the MySQL shell, run the following commands:

```
CREATE DATABASE mydatabase;

CREATE USER 'dbuser'@'localhost' IDENTIFIED BY 'SecurePass123!';

GRANT ALL PRIVILEGES ON mydatabase.* TO 'dbuser'@'localhost';

FLUSH PRIVILEGES;
```

Explanation:

CREATE DATABASE mydatabase;: Creates a new database named mydatabase.

CREATE USER 'myuser'@'localhost' IDENTIFIED BY 'mypassword';: Creates a new user myuser with the password mypassword.

GRANT ALL PRIVILEGES ON mydatabase.* TO 'myuser'@'localhost';: Grants all privileges on mydatabase to myuser.

FLUSH PRIVILEGES;: Reloads the privilege tables.

EXIT;: Exits the MySQL shell.

4) Create a PHP Script to Connect to MySQL

Command:

sudo nano /var/www/html/testdb.php

sudo chmod 644 /var/www/html/testdb.php

Explanation:

PHP Script Example

Filename: testdb.php

4) Create a PHP Script to Connect to MySQL

```
<?php
// Enable error reporting
error_reporting(E_ALL);
ini_set('display_errors', 1);

// Database credentials
$servername = "localhost";
$username = "dbuser";
$password = "SecurePass123!";
$dbname = "mydatabase";

// Create connection
$conn = new mysqli($servername, $username,
$password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
```

Explanation:

Save the file and exit the editor (Ctrl + X, then Y, then Enter).

This PHP script connects to the MySQL database using the provided credentials and checks if the connection is successful.

5) Test the PHP Script

Open a web browser.

Navigate to http://localhost/testdb.php

Expected Result:

The browser should display "Connected successfully" if the connection is successful.

Additional Tips

1) Restart Apache

Command:

sudo systemctl restart apache2

2) Verify PHP Installation

Command:

php -v

Removing and Cleaning the Installations

Removing Apache:
sudo systemctl stop apache2

sudo apt-get purge apache2 apache2-utils
apache2-bin apache2.2-common

sudo rm -rf /etc/apache2 /var/www/html
/var/log/apache2

Removing and Cleaning the Installations

Removing MySQL:

```
sudo systemctl stop mysql

sudo apt-get purge mysql-server mysql-client
mysql-common mysql-server-core-*
mysql-client-core-*

sudo rm -rf /etc/mysql /var/lib/mysql /var/log/mysql
```

Removing and Cleaning the Installations

Removing PHP:

sudo apt-get purge php*

sudo rm -rf /etc/php /var/lib/php /var/log/php

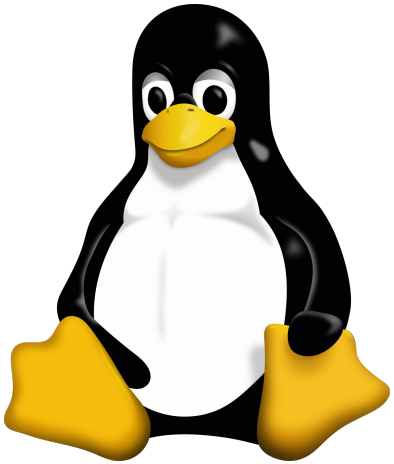
Removing and Cleaning the Installations

Update package lists and remove unnecessary packages:

```
sudo apt-get update
sudo apt-get autoremove
sudo apt-get autoclean
```

Q/A Session

Thank you !



End of Day 5!

By Maya Mnaizel