TASK

Additional Reading - Installation

Visit our website

INTRODUCTION

If you would like to install additional third-party packages you should be aware of best practices relating to installation. The use of a package manager is highly recommended and containerisation is important when different projects or applications require specific requirements.

PACKAGE MANAGERS

Package managers provide a means to install packages, organise them in your file system, and manage dependencies between packages. A dependency is when a package depends on another particular version of another package to function. If you have two different packages that rely on different versions of another package this creates problems if you don't have a package manager. It can also be very time-consuming to find and install all the dependencies for a package. Package managers also check for known vulnerabilities that may pose a security risk. You can use a package manager to share packages you have created with others.

Two types of package managers: those for **operating systems** and those for **programming languages**. Package managers are linked to software repositories where the packages or software are stored.

Some common operating system package managers include:

- **Chocolatey** for Windows (**Chocolatey repository**)
- **HomeBrew** for MacOS (**Homebrew repository**)
- Linux – each distribution has its own package manager
  - **DNF** for Fedora
  - **APT** for Ubuntu

As mentioned, programming languages typically have language-specific package managers. Examples of some programming languages and commonly used package managers are summarised in the table below.

Language	Package Manager	Software Repository
Python	pip	pypi
Python	conda	Anaconda Packages
Java	Maven	Maven Central
Java	Gradle	Maven Central
JavaScript	npm	npm

For Python, the general purpose package manager is pip whereas conda is aimed at data science programming. The package manager you use depends on your needs and preferences.

Extra resource

You can find many Python libraries and packages in the **pypi** or **Anaconda** software repository.

CONTAINERISATION

As you start working on software applications and larger projects that need to be deployed, you will also need to consider how to bundle your code with all the third-party packages it relies on.

If you are using Python the main containerisation method is the Python venv module. Explore the **venv module** documentation for more information. If you are using Anaconda you can create a **conda virtual environment** to package your Data Science projects. **Docker** is a more general method of containerisation and supports several programming languages.

You will not need to use any of these containerisation methods in this bootcamp, but it is best practice to use a container for individual applications or projects.

To install the packages you will need to become familiar with the command line.

THE COMMAND LINE

The command line is a means of interacting with a computer program where the user issues commands to the program in the form of successive lines of text. With the command line, you can quickly issue instructions to your computer, getting it to do precisely what you want it to do. The command line is rarely used by most end users since the advent of the Graphical User Interface (a more visual way of interacting with a computer using items such as windows, icons, menus, etc.). However, you will find it helpful to use the command line when installing third-party packages and other actions such as version control with Git.

Finding The Command Line

In Windows, you can simply click the Start menu and type `powershell` in the search box to locate the command line. Alternatively, the command line should be one of the options under 'Programs' and you can simply click on the application to open it.

With macOS, open the command line by opening the terminal. This can be done by opening the Applications folder, navigating to Utilities and then launching Terminal. Alternatively, you can search for "terminal" to find the application to launch.

Take note:

You will need to ensure that you have **administrator access** on Windows or superuser access on macOS, Fedora or Ubuntu to install packages.