DAY 4

Logging, Managing packages with pip, and Floating point Arithmetic.

Logging:

Logging is a process of recording events, messages, or information about the execution of a program. It's a valuable tool for debugging and understanding the flow of a program. In Python, the logging module provides a flexible framework for emitting log messages from Python programs. You can use it to output messages to different places (console, files) at different levels (debug, info, warning, error, critical).

Managing Packages with pip:

pip is the package installer for Python. Inside a virtual environment, you can use pip to install, upgrade, and uninstall packages. You can install, upgrade, and remove packages using a program called pip. By default pip will install packages from the Python Package Index, <https://pypi.org>. You can browse the Python Package Index by going to it in your web browser, or you can use pip’s limited search feature.

Virtual Environments:

A virtual environment is an isolated Python environment where you can install packages and dependencies without affecting the global Python installation. This is useful to manage project-specific dependencies and versions. Different applications can then use different virtual environments. To resolve the earlier example of conflicting requirements, application A can have its own virtual environment with version 1.0 installed while application B has another virtual environment with version 2.0. If application B requires a library be upgraded to version 3.0, this will not affect application A’s environment.

Creating Virtual Environments:

To create a virtual environment, you can use the venv module (available in Python 3.3 and newer) or virtualenv (a third-party tool). Here's an example using venv. To create a virtual environment in Python, you can use the venv module, which is included in Python 3.3 and newer versions.

Floating Point Arithmetic:

Floating-point arithmetic is a way to represent and perform arithmetic with real numbers (those that have a fractional part). In Python, floating-point numbers are represented as float data type. Floating-point numbers are represented in computer hardware as base 2 (binary) fractions. For example, the decimal fraction 0.125 has value 1/10 + 2/100 + 5/1000, and in the same way the binary fraction 0.001 has value 0/2 + 0/4 + 1/8. These two fractions have identical values, the only real difference being that the first is written in base 10 fractional notation, and the second in base 2.