

ch1_the_linux_foundation

June 27, 2021

1 Chapter 1 - The Linux Foundation

1.1 Introduction and Learning Objectives

By the end, you should be able to:

- Discuss the role of the Linux Foundation.
 - Appreciate the learning opportunities provided by the TLFs training program.
 - Describe the software environment required for the course.
 - Describe the three major Linux distribution families.
-

1.2 The Linux Foundation

- Everything in python is an **object** and has a type.
 - Objects are a **data abstraction** that capture:
 - Internal representation through data attributes.
 - Interface for interacting with object through methods (procedures), defines behaviors but hides implementation.
 - Can **create new instances of objects**.
 - Can also **destroy objects**.
-

Built in data objects:

- Lists.
- Tuples.
- Strings.

We want to explore ability to create our own data object types.

How are lists represented internally? They are a linked list of cells. But it doesn't matter. What matters is how to **manipulate** them. How do you do that?

- L[i], L[i:j], L[i:j,k]
- len(), min(), max(), etc.

Internal representation should be **private**. Correct behavior may be compromised if you manipulate it directly - instead you use **defined interfaces** to do so.

How to create your own objects with classes?

- Distinction: creating a class and using an instance of said class.
 - A list is a class. L1 = [1,2,3] is an instance of said class.
 - Creating the class involves:
 - Defining the class name.
 - Defining class attributes.
 - Using the class involves:
 - Creating new instances of objects.
 - Doing operations on the instances.
-

Advantages of OOP

- Bundle data into packages together with procedures that work on them through well-defined interfaces.
- Divide-and-conquer development.
 - Implement and test behavior of each class separately.
 - Increased modularity reduces complexity.
- Classes make it easy to reuse code.
 - Many Python modules define new classes.
 - Each class has a separate environment (no collision on function names).
 - Inheritance allows subclasses to redefine or extend a selected subset of a superclass' behavior.