В библиотеке random механизм генерации случайных чисел основан на использовании псевдослучайных чисел. Внутри библиотеки есть генератор случайных чисел, который инициализируется начальным зерном (seed) и использует различные алгоритмы для генерации последовательности чисел, которые кажутся случайными. Генераторы псевдослучайных чисел в random обычно основаны на алгоритме Мерсенна-Твистера.  
  
Функциональные объекты отличаются от функций и лямбда-выражений тем, что они представляют собой объекты, которые могут быть вызваны как функции. Функциональные объекты могут быть созданы с использованием классов, которые реализуют специальные методы \_\_call\_\_ или \_\_invoke, или с использованием функциональных объектов из стандартной библиотеки Python, таких как functools.partial. Функции и лямбда-выражения, с другой стороны, являются блоками кода, которые могут быть вызваны и выполнены.  
  
Алгоритмам стандартной библиотеки требуются итераторы вставки в случаях, когда они должны модифицировать содержимое контейнера, например, добавить элементы или удалить существующие элементы. Итераторы вставки предоставляют абстрактный интерфейс для вставки элементов в контейнеры. Это позволяет алгоритмам стандартной библиотеки работать с различными типами контейнеров, не завися от их конкретной реализации.  
  
Классификация алгоритмов стандартной библиотеки обычно основывается на их функциональности. В стандартной библиотеке Python предлагается несколько классификаций, таких как алгоритмы сортировки (например, sorted), алгоритмы поиска (например, in оператор), алгоритмы преобразования и манипулирования последовательностями (например, map, filter, reduce), алгоритмы работы с итераторами (например, zip, enumerate) и другие.  
  
Алгоритмы стандартной библиотеки предпочтительнее собственных циклов, потому что они обычно предоставляют более высокий уровень абстракции и упрощают код. Алгоритмы стандартной библиотеки часто оптимизированы и хорошо протестированы, что делает их надежными и эффективными. Использование готовых алгоритмов также способствует повышению читаемости и поддерживаемости кода, поскольку они предлагают стандартизированный подход к решению общих задач.