**Python DB-API** – это не конкретная библиотека, а набор правил, которым подчиняются отдельные модули, реализующие работу с конкретными базами данных. Отдельные нюансы реализации для разных баз могут отличаться, но общие принципы позволяют использовать один и тот же подход при работе с разными базами данных.

DB API 2.0 регламентирует интерфейсы модуля расширения для работы с базой данных, методы объекта-соединения с базой, объекта-курсора текущей обрабатываемой записи, объектов различных для типов данных и их конструкторов, а также содержит рекомендации для разработчиков по реализации модулей. На сегодня Python поддерживает через модули расширения многие известные базы данных (уточнить можно на web-странице по этому адресу). Ниже рассматриваются почти все положения DB-API за исключением рекомендаций для разработчиков новых модулей.

Для кождой бази даных есть свой DB-API модуль

С помощю DB-API можно устанавливать соединение с базой, чтение из базы, запись в базу и т.д.

Здесь необходимо сказать о том, что должен предоставлять модуль для удовлетворения требований DB-API 2.0. Доступ к базе данных осуществляется с помощью объекта-соединения (connection object). DB-API-совместимый модуль должен предоставлять функциюконструктор connect() для класса объектов-соединений. Конструктор должен иметь следующие именованные параметры:

• dsn Название источника данных в виде строки

• user Имя пользователя

• password Пароль

• host Адрес хоста

• database Имя базы данных.

import random

lst = []

for \_ in range(x):

    input\_number = int(input('generate number between 0 and 1023 '))

    appnd\_number = random.randint(0, 1023)

    lst.append(appnd\_number)

    if appnd\_number == 0:

        break