Ответы на вопросы:

1. В итерируемых объектах. Yield используется как аналог return. Но он возвращает генератор позволяя сохранить состояние функции, а потом вернуться и начать с того же места. Программа будет выполняться до конца, если не встретит ещё раз инструкцию yield.
2. Декораторы – это обёртки функций или классов перехватывающие выполнение функции или попытки создания экземпляров, позволяющие добавить автоматически вызываемый программный код. Конкретный и явный синтаксис позволяет оптимизировать программный код улучшая его читаемость. Декораторы применяются к функции или к классу только один раз и не нужно добавлять дополнительный код. Декораторы способствуют инкапсуляции и это тоже уменьшает избыточность кода и позволяет упростить внесение изменений в будущем.
3. GIL предназначен для сериализации доступа к внутренним компонентам интерпретатора из разных потоков. В любой момент времени работает только тот поток, который содержит в себе интерпретатор. Поток получает GIL, выполняет небольшую работу, а затем передает GIL следующему потоку.
4. Выведет синтаксическую ошибку в строке **print** argument (отсутствуют скобки). Если скобки поставить, то выведет значение – 5.

5

5

5

5

5

5

1. the\_tuple\_1 **=** **()** Синтаксически литерал кортежа заключается в круглые скобки.

Данный код имеет вложенный цикл. Программа запустит внешний и в первой его итерации перейдет во вложенный. Затем она снова вернется к началу внешнего и снова вызовет внутренний. Это будет происходить до тех пор, пока последовательность не завершится. Lambda : I включает все элементы исходной последовательности.

1 1 1

1 2 1

3 2 3

Мы создали классы и поставили две заглушки, предполагая, что потом будем добавлять какие-то действия. Классы **Child1 и** **Child2 наследуют класс Parent, а, следовательно, и значение переменной х = 1. Выводим это на экран. Затем классу Child1 записываем значение х = 2, снова выводим результат на экран. Потом классу Parent записываем значение х = 3, которое так же наследуется классом Child2 и выводим результат на экран.**

1. Библиотека Celery, и два брокера: RabbitMQ, AMQP.
2. Задачи в веб-приложениях: анализ данных, экспорт документов, сканирование информации из разных источников, рассылка сообщений.
3. Для создания сетевого соединения нам нужно импортировать соответствующий модуль. import socket Затем нужно создать сам сокет sock = socket.socket()

AF\_INET – это сокет, который будет работать с адресами, соответствующими, адресам IPv4

SOCK\_STREAM – это протокол транспортного уровня, который соответствует протоколу TCP Для UPD так же AF\_INET, он соответствует адресам IPv4

Для протокола - SOCK\_DGRAM, он соответствует протоколу UDP

1. Точка обмена в RabbitMQ – это посредник между отправителем и получателем. Она нужна для независимости между ними. Поступающие сообщения обрабатываются в точке обмена и затем распределяются согласно внутренним правилам.
2. ORM используется тогда, когда нам необходимо запрашивать и обрабатывать данные из базы данных с использованием объектно-ориентированной парадигмы. Пример: у нас есть класс книги и нам нужно получать книги, автором которых является известный нам писатель.