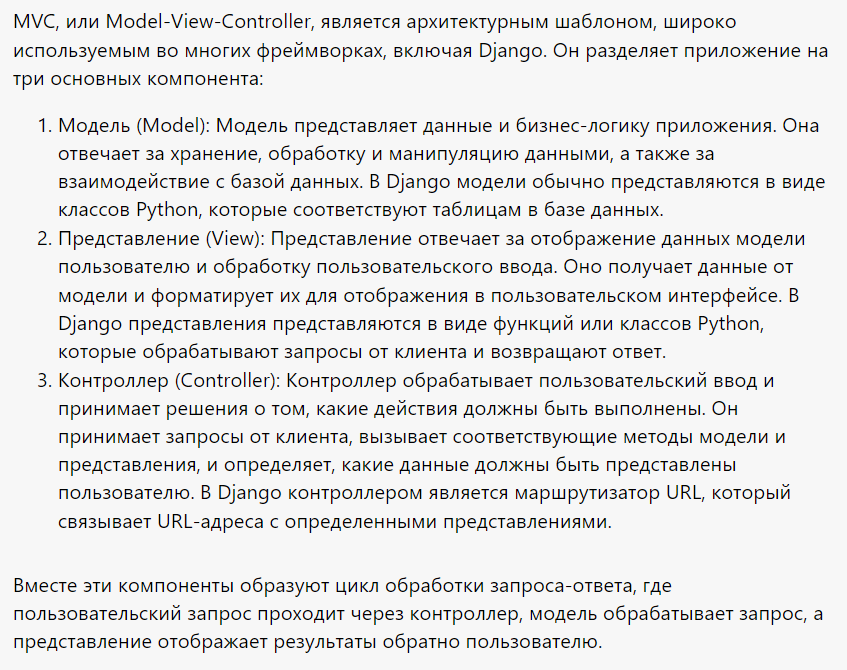
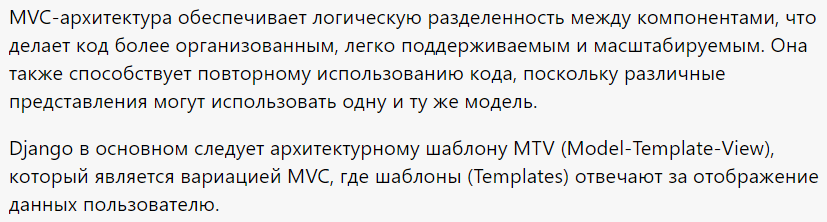
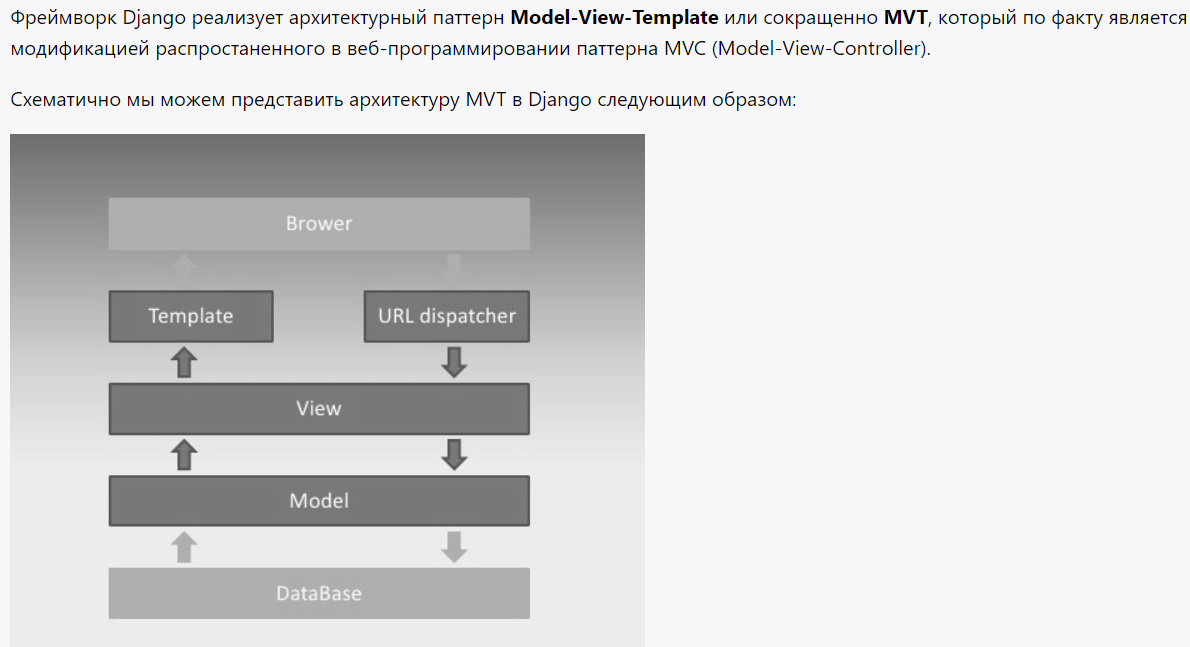
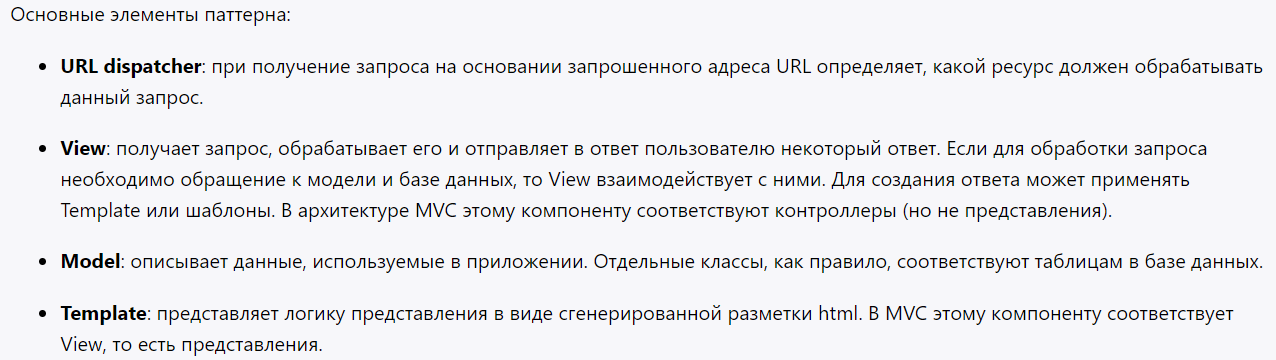


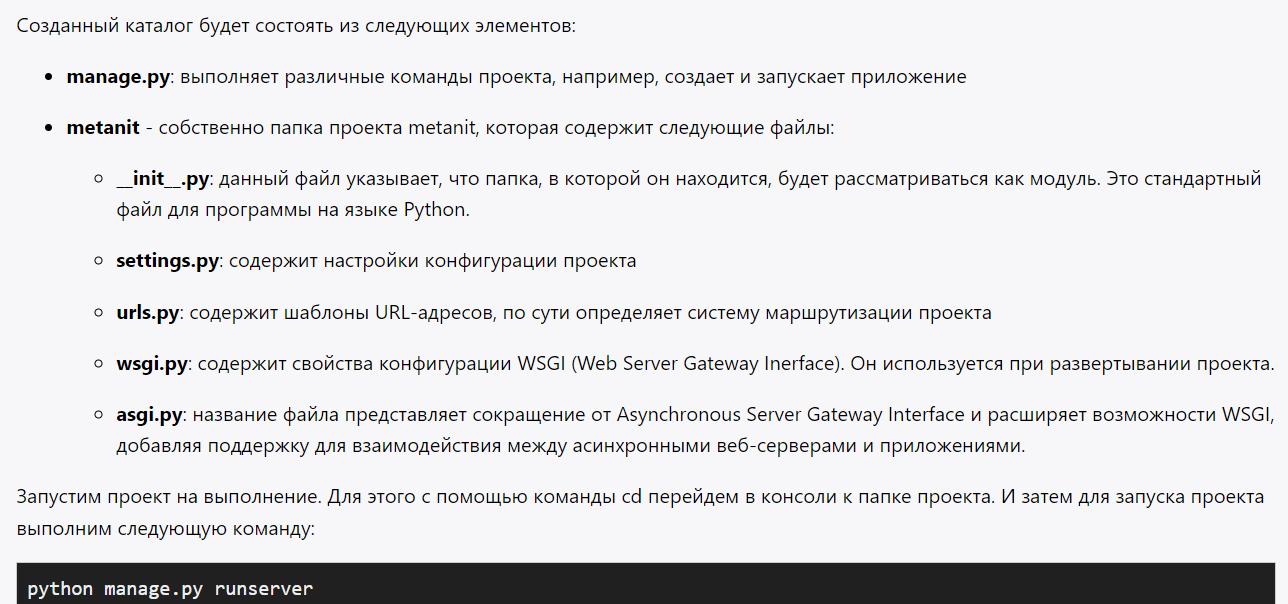
Реализовано с помощью MVC  
  


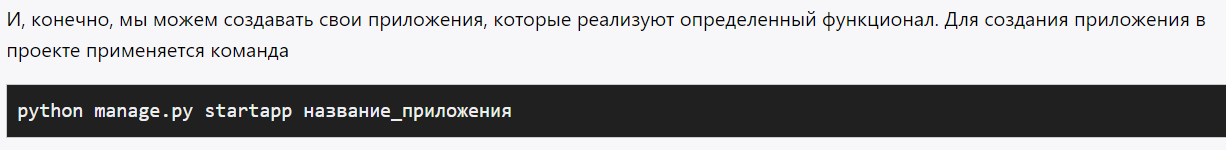




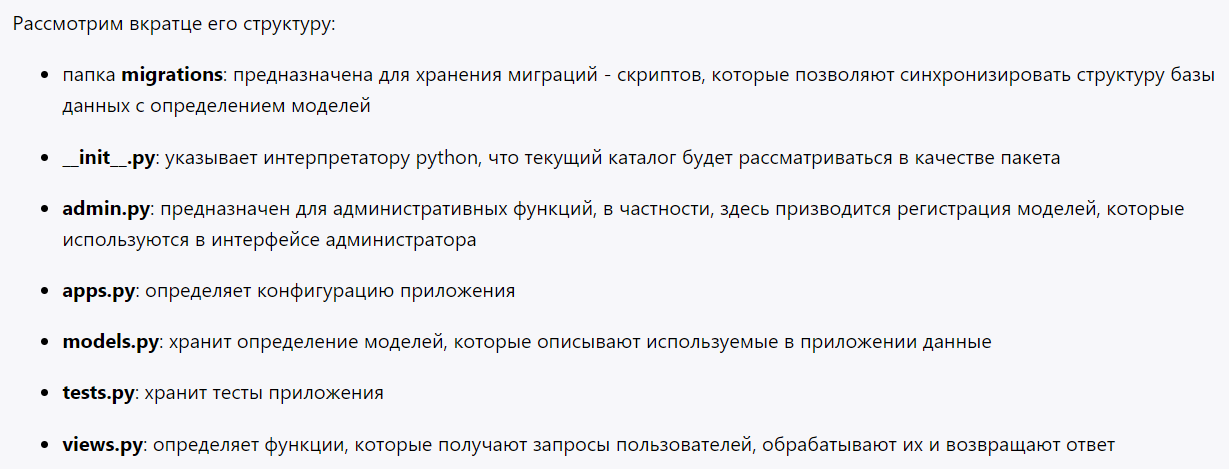
Когда к приложению приходит запрос, то URL dispatcher определяет, с каким ресурсом сопоставляется данный запрос и передает этот запрос выбранному ресурсу. Ресурс фактически представляет функцию или View, который получает запрос и определенным образом обрабатывает его. В процессе обработки View может обращаться к моделям и базе данных, получать из нее данные, или, наоборот, сохранять в нее данные. Результат обработки запроса отправляется обратно, и этот результат пользователь видит в своем браузере. Как правило, результат обработки запроса представляет сгенерированный html-код, для генерации которого применяются шаблоны (Template).

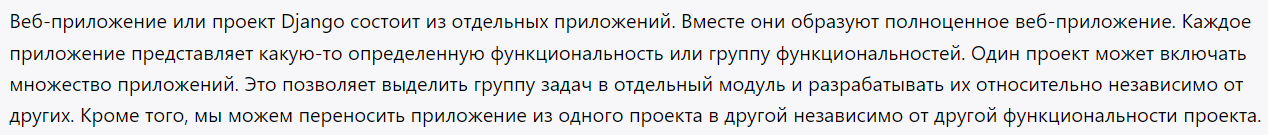
Django

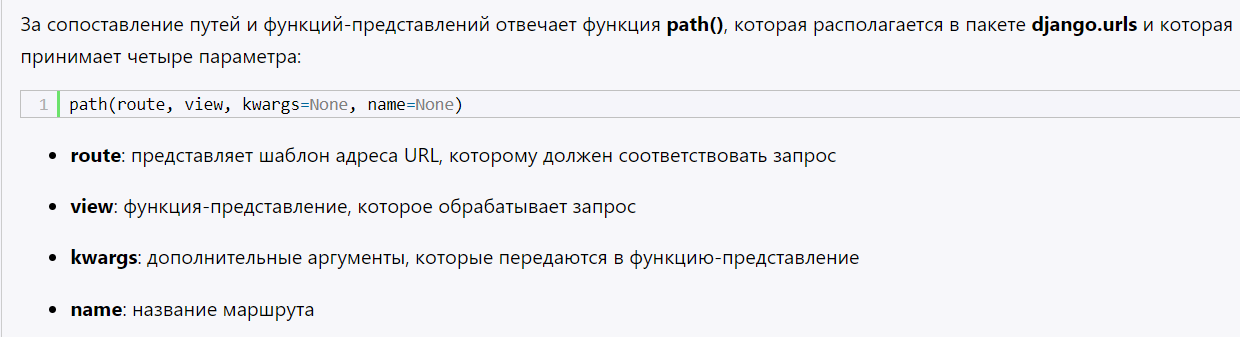


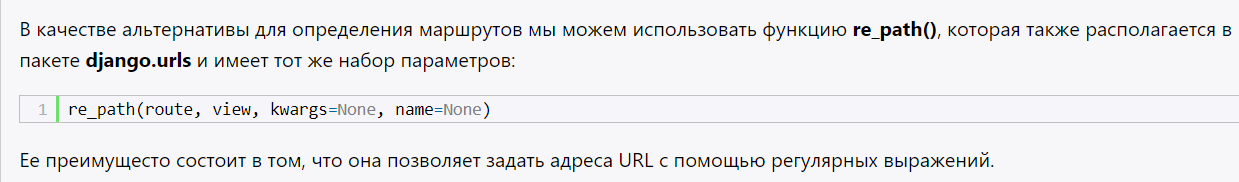


Cтруктура приложения



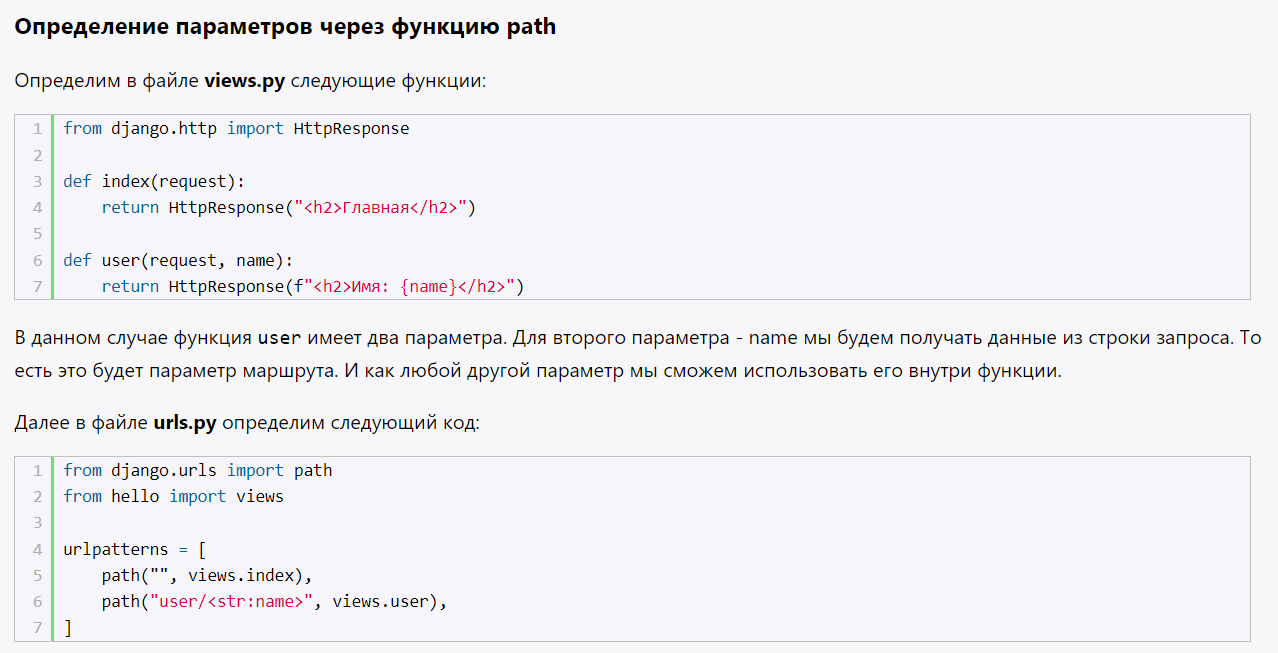


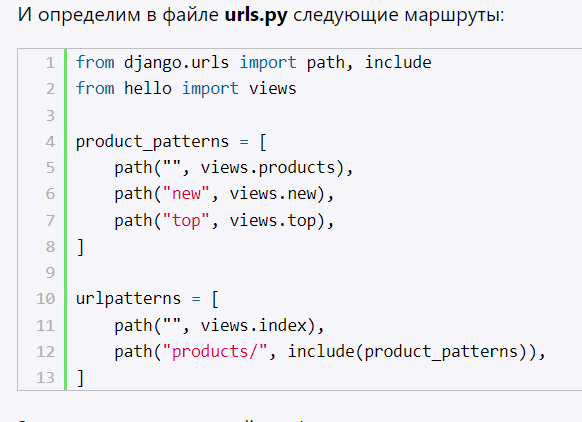


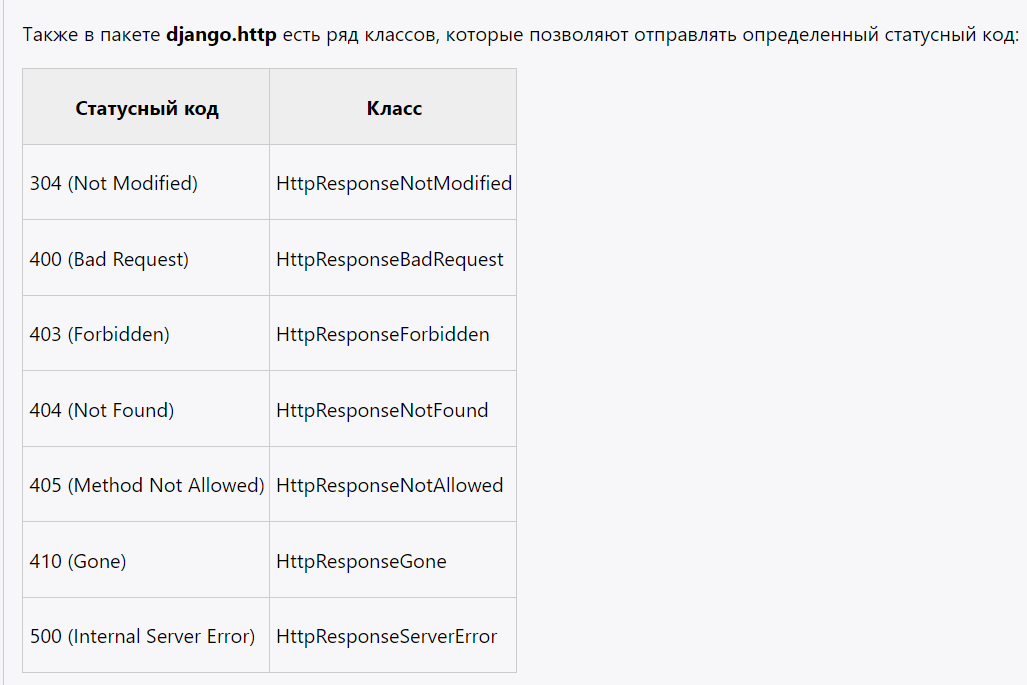


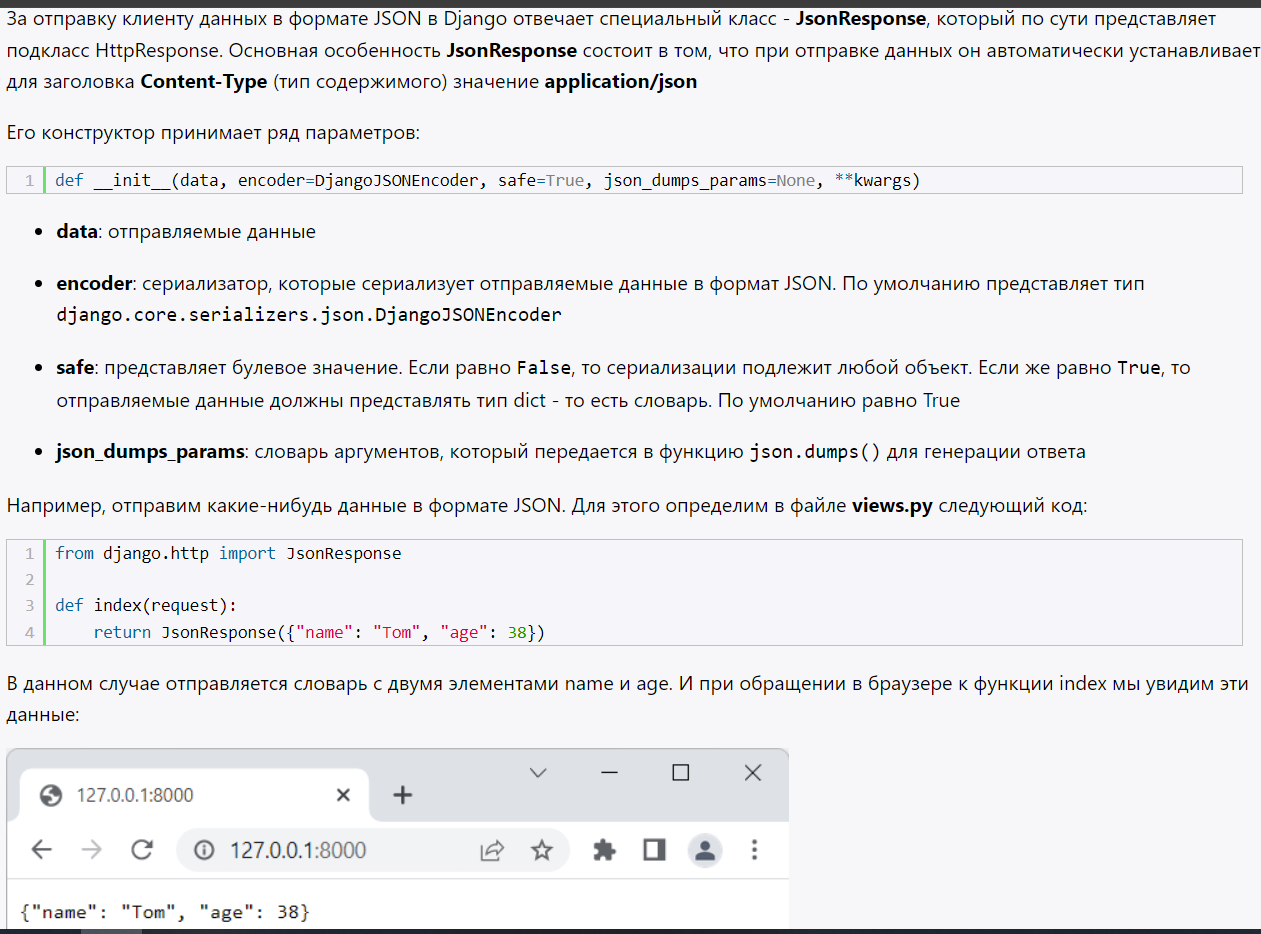
РАБОТА С HTTP REQUEST <https://metanit.com/python/django/3.7.php>

РАБОТА С HTTP RESPONSE <https://metanit.com/python/django/3.8.php>

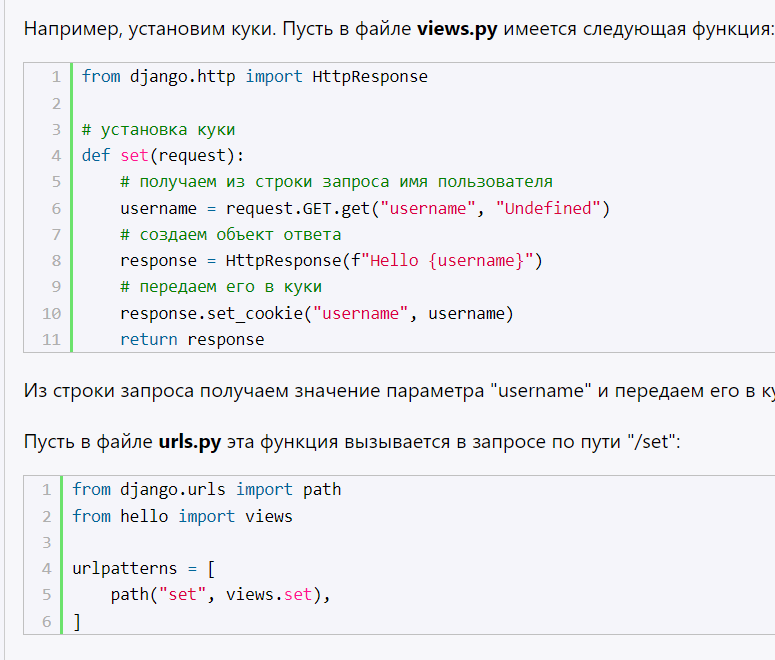


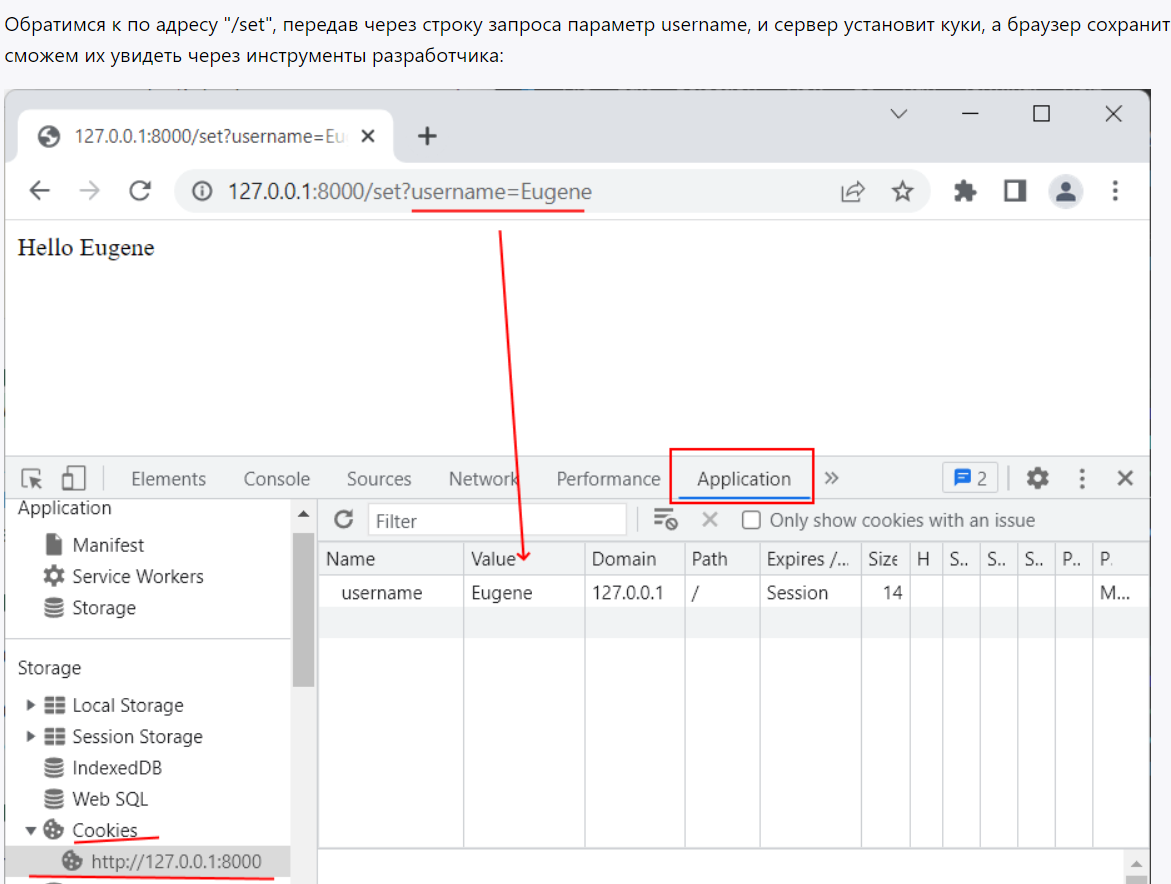


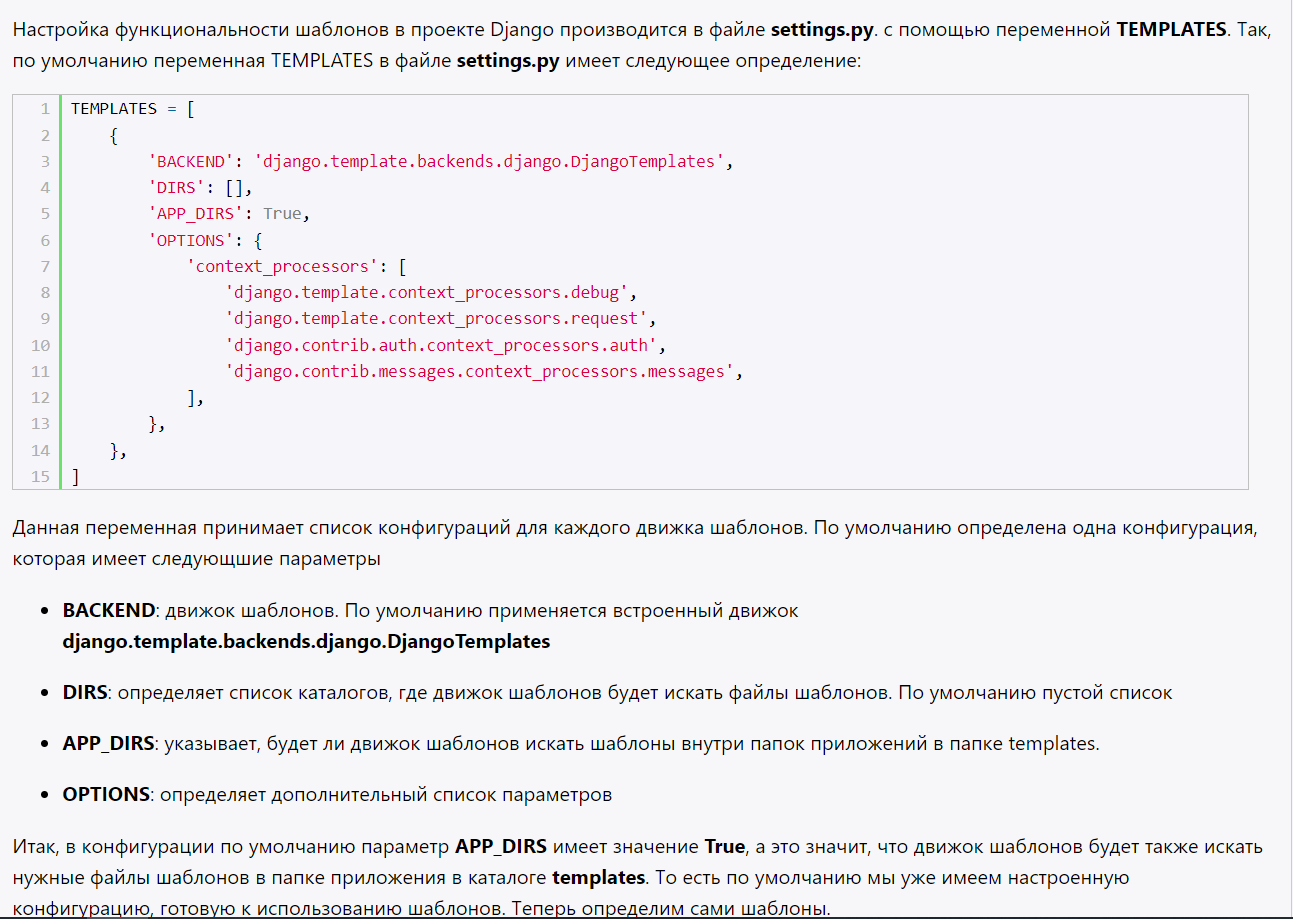


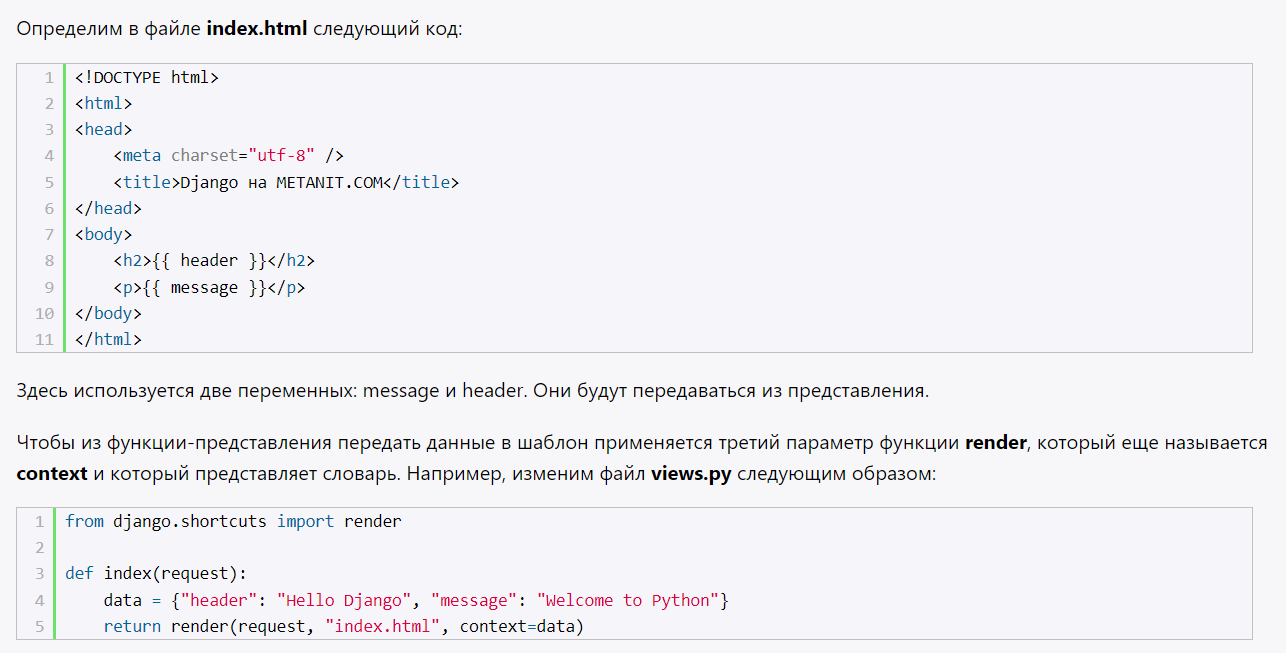


Куки представляют самый простой способ сохранить данные пользователя. Куки хранятся на компьютере пользователя и могут устанавливаться как на сервере, так и на клиенте. Так как куки посылаются с каждым запросом на сервер, то их максимальный размер ограничен 4096 байтами.





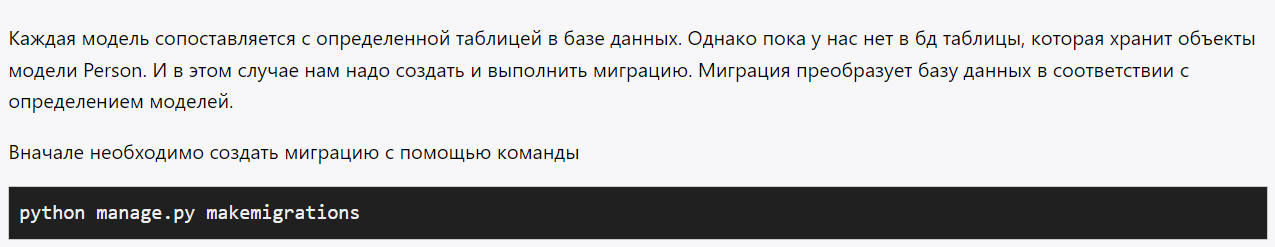


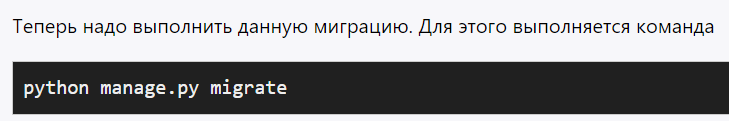
ПЕРЕДАЧА ПАРАМЕТРОВ В ШАБЛОН  


render = TemplateResponse

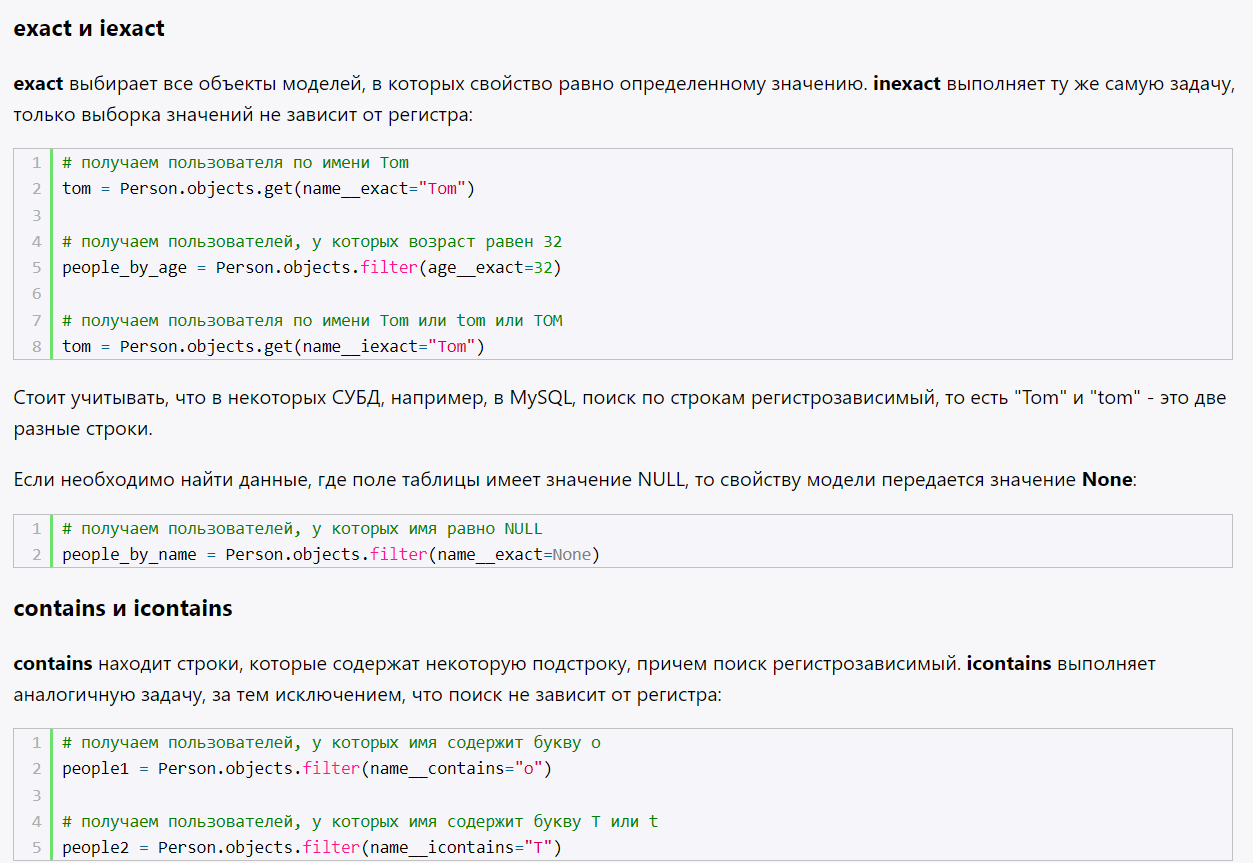


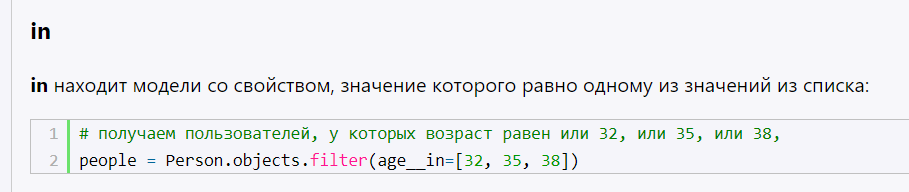
<https://metanit.com/python/django/2.5.php>



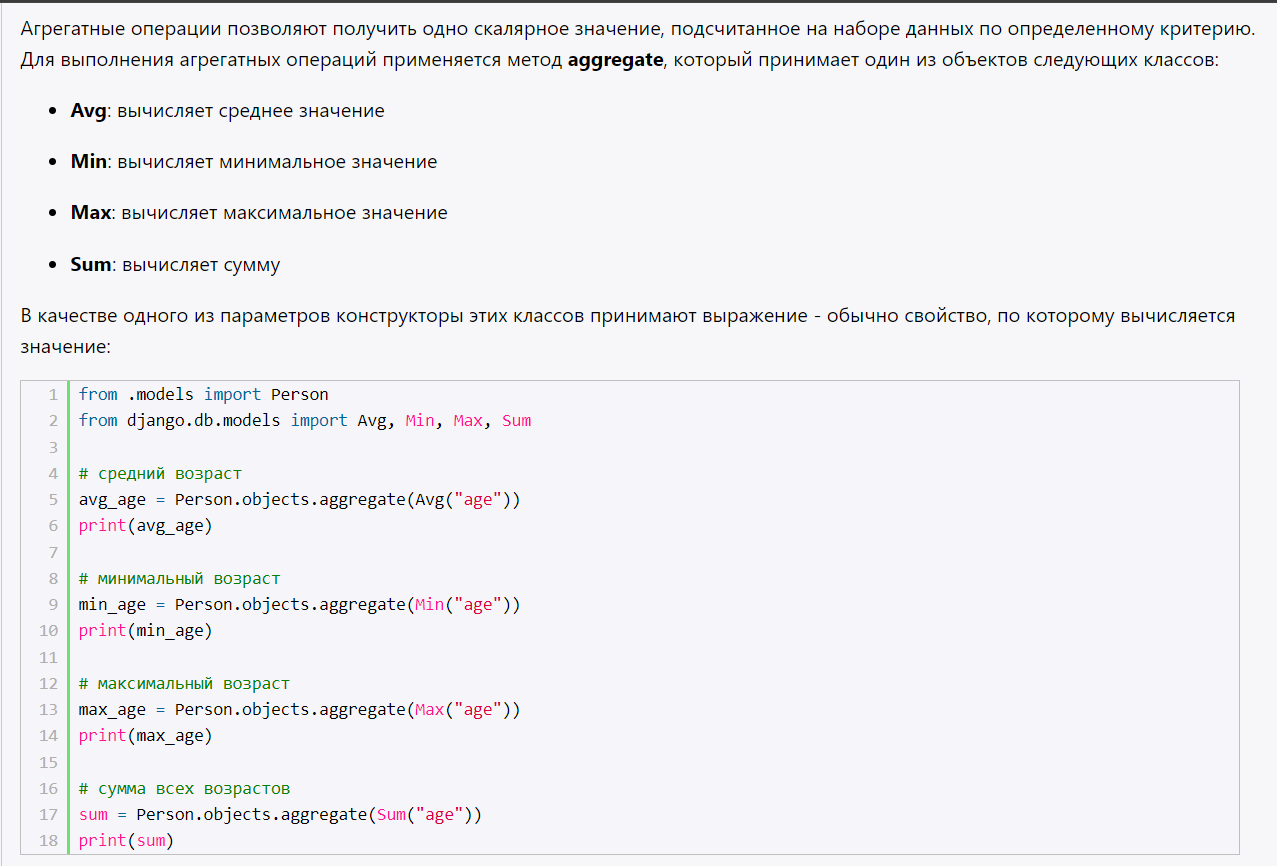


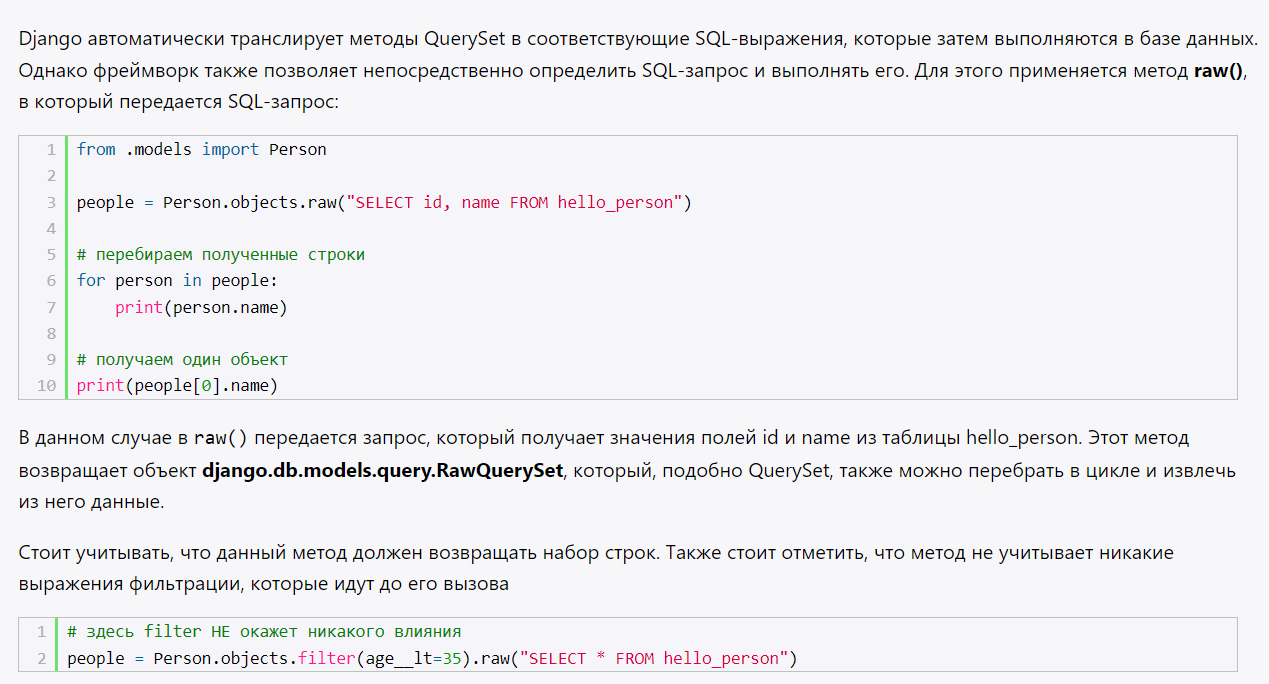
python manage.py createsuperuser – создать админа чтобы зайти на сайт

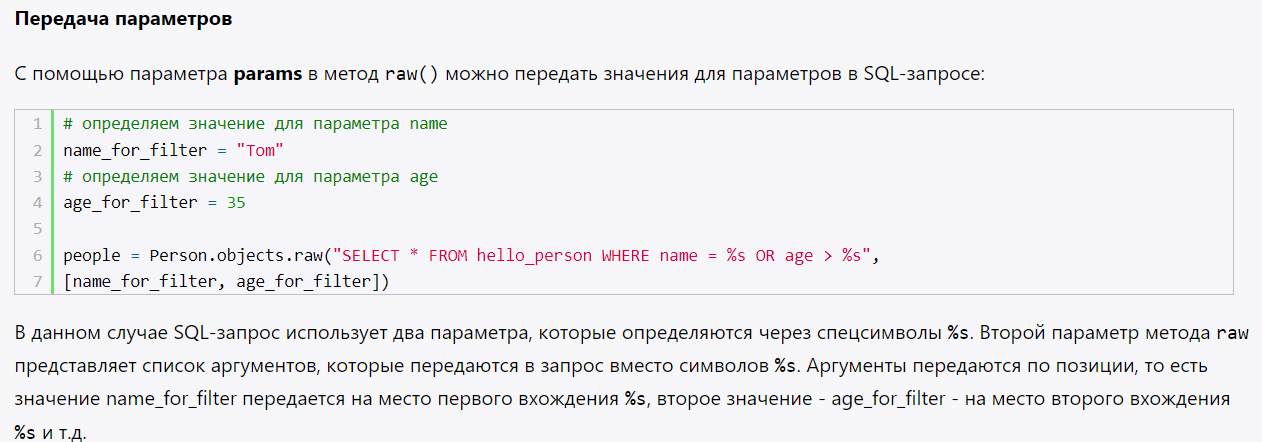


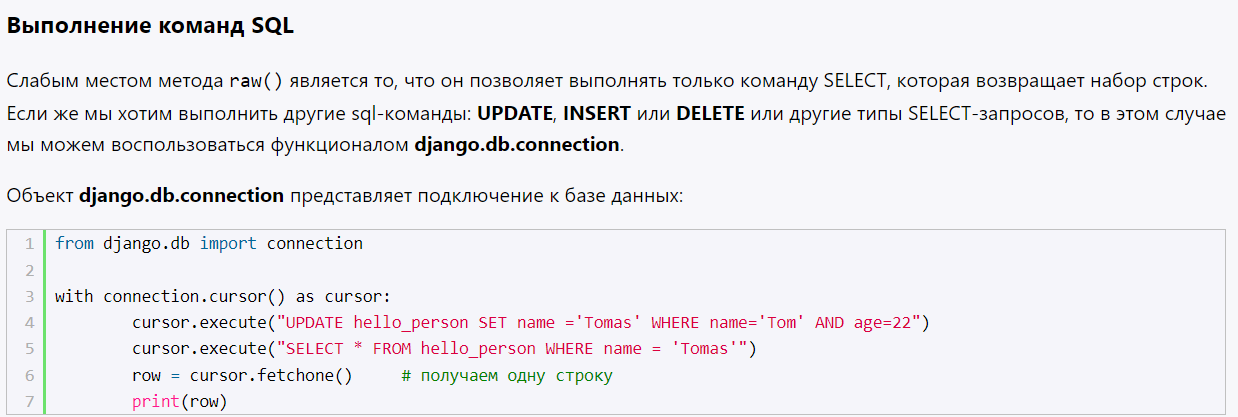


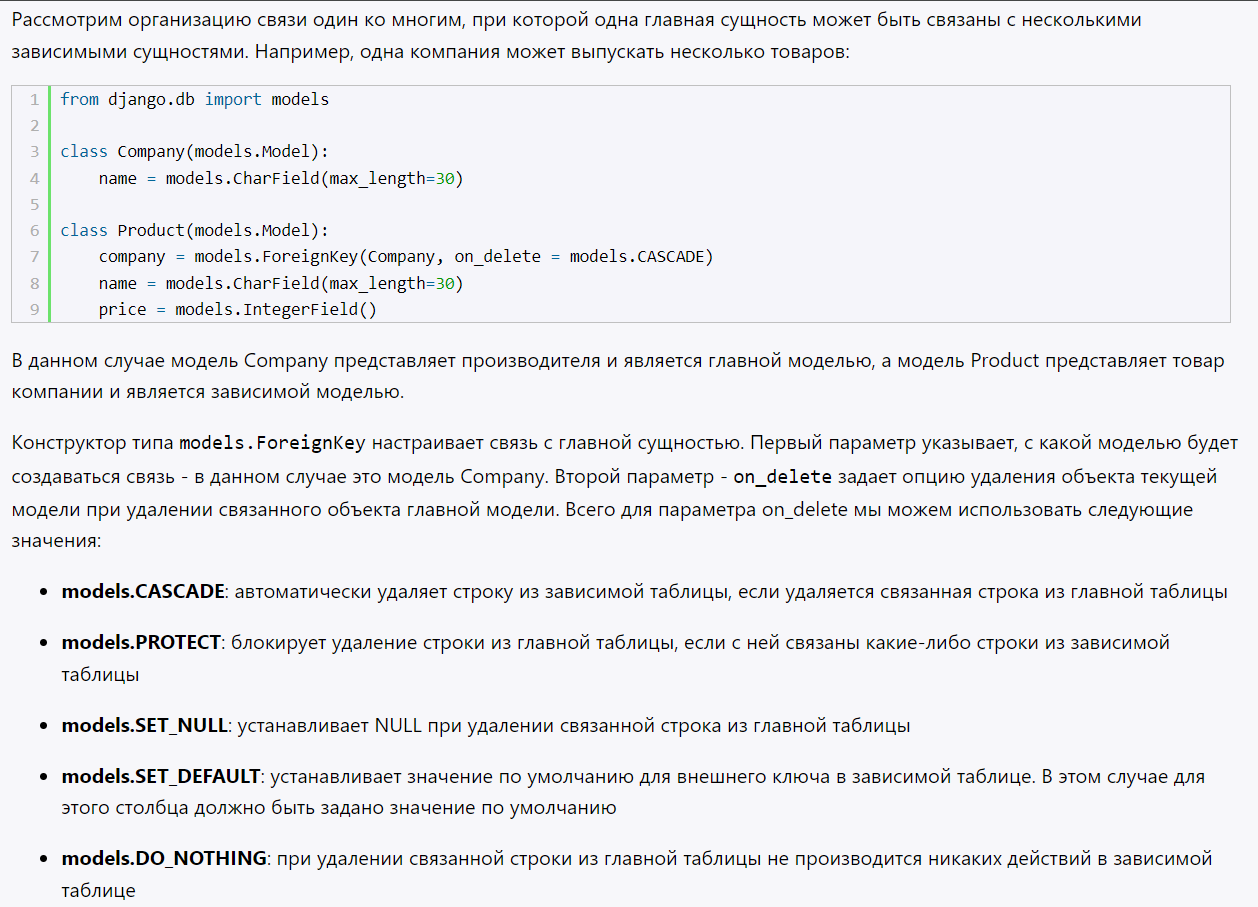
И т.д (все такое <https://metanit.com/python/django/5.13.php> тут)



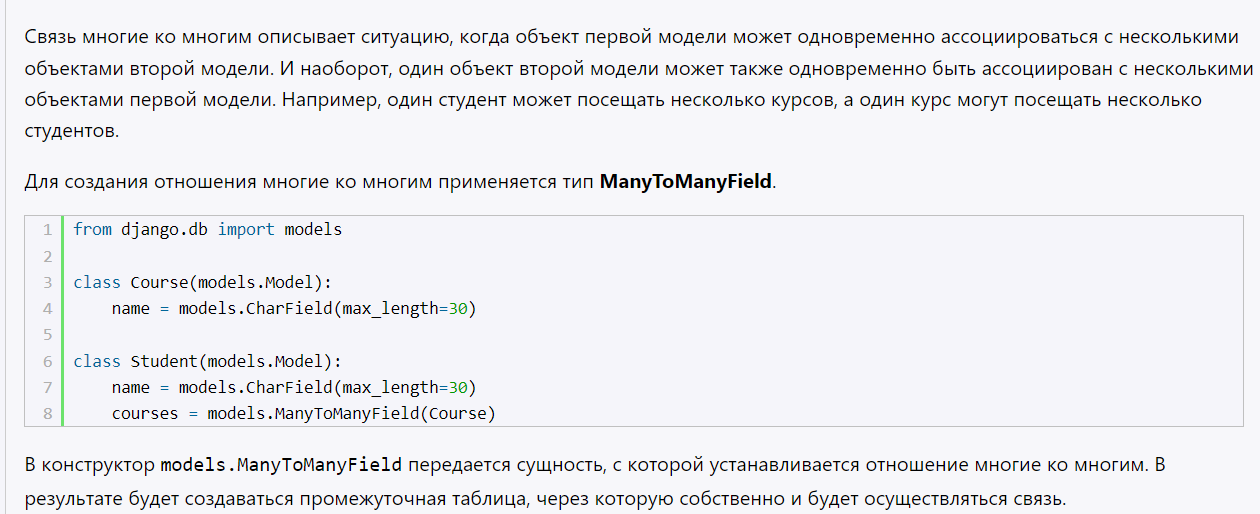




  
rows = cursor.fetchall() # получаем все строки



Операции с этим тут <https://metanit.com/python/django/5.6.php>



Ответы на теориму в лабе

