

# [Community] Switch PHP to Python Guide

1. Первый этап свитча - это ознакомиться с синтаксисом питона, его структурами данных и т. д. Всё это хорошо описано в небезызвестном Лутце: <http://nqnwebs.com/IMG/pdf/OREilly-Learning-Python-4th-Edition-Oct-2009.pdf>
2. Далее, чтобы отработать синтаксис питона и работу с ним в нескучной форме предлагается пройти задания на <http://www.pythonchallenge.com/>.
3. Теперь самое время изучить основные стайл гайды по питону и желательно выучить их наизусть: <https://www.python.org/dev/peps/pep-0008> - стандартный стайл гайд, а вот так <https://google-styleguide.googlecode.com/svn/trunk/pyguide.html> принято на питоне писать в гугле
4. Далее must have в коллекции прочитанных книг по питону Cookbook [http://files.grouper.org/it-ebooks/python\\_cookbook\\_2nd\\_edition.pdf](http://files.grouper.org/it-ebooks/python_cookbook_2nd_edition.pdf). Просто отличнейшая книга в интереснейшем формате: есть проблема, которая может стать перед питон разработчиком, есть пример решения, как это сделать правильно, используя все возможности питона. Очень-очень советую эту книгу прочитать!
5. После изучения питона как такового следует начинать читать про web фреймворки под питон. Ну и, конечно, начать стоит с джанги: <https://www.djangoproject.com/>. Ничего лучше, чем документация джанги, про джангу не расскажет (ну возможно только, если исходники её смотреть, но это уже какой-то следующий этап). Обратит внимание на Models раздел, работу с ORM и т. д. Посмотреть, почитать про Class Based Views. Ну и многое другое.
6. После того, как изучили джангу, следует почитать про то, как её правильно использовать: <http://effectivedjango.com/index.html>. Не стоит думать, что это какая-то недодокументация по джанге. На самом деле на этом ресурсе описаны best practices по работе с django, которые надо знать. Из ключевых вы тут найдёте: как грамотно использовать ORM, дабы не ложить базу большим количеством запросов, как грамотно использовать те же CBV. Ресурс небольшой, но на многие вопросы даёт ответы.
7. Ещё одна отличная книга по эффективной работе с джанго: Daniel Greenfeld - Two Scoops of Django - 2013.pdf. Здесь вы сможете найти ещё больше best practices по работе с джангой, чем в предыдущем пункте.
8. Теперь самое время посмотреть на какие-то мелкие приложения на джанго, например тут: <https://github.com/nathanborror/django-basic-apps>. Стоит полазить по этому репо, посмотреть, как описывать модели, админки, вьюшки для простых прикладных задач.
9. Вот и настало время написания первого приложения на django. Первое приложение, как бы это банально ни было, будет секс-шоп. Требования к секс-шопу:
  - a. Основные сущности:
    - i. товар, имеет название, описание, стоимость, категорию, производителя, количество на складе
    - ii. производитель, имеет название, страну
    - iii. категория, имеет название
    - iv. новость, имеет название, описание, дату публикации
  - b. Основные роли:
    - i. админ, имеет доступ в админку и может добавлять/изменять/удалять товары, менять их цену и количество на складе, может добавлять/изменять/удалять производителей и категории, может добавлять/изменять/удалять новости
    - ii. пользователь, может просматривать только новости, товары. Если залогинен, то ещё может добавлять/удалять товары в/из корзину, оформлять покупку
  - c. Основные экраны:
    - i. главная страница, на главной отображаются последние 5 новостей и пагинация для них, есть ссылка на категории товаров и, если пользователь залогинен, корзину. Если новостей нет, то ни они, ни пагинация не отображаются
    - ii. страница категорий товаров, тут отображаются категории, в которых есть товары, и ссылки на эти категории. Категория отображается, если хотя бы один товар из неё есть на складе.
    - iii. страница категории товаров, тут отображаются товары категории, которые есть на складе. У каждого товара отображается его стоимость и количество на складе. Если пользователь залогинен, то есть кнопка возле товара "Добавить в Корзину". По её нажатию товар добавляется в корзину и после этого может быть оформлен заказ.
  - d. Страница оформления заказа:
    - i. тут отображаются все товары, которые были добавлены в корзину и кнопка оформить заказ, по её нажатию пользователю отправляется письмо, в котором написано, какой он шалун.
    - ii. если товаров в корзине нет, то пользователю отображаются сообщение "У вас ещё нет товаров в корзине" и последние 5 просмотренных им товаров с кнопками "Добавить в корзину"
    - iii. если пользователь не смотрел ещё ни одного товара, то ему отображается сообщение "У вас ещё нет товаров в корзине" и ссылка на страницу категорий товаров
10. Спускаемся ближе к WG реалиям. У нас мы используем ряд питоновских пакетов/django батареек/приложений, которые упрощают работу и/или расширяют возможности django. Поэтому стоит почитать про:
  - a. <https://south.readthedocs.org/en/latest/> - миграции для джанго моделей
  - b. <http://jinja.pocoo.org/docs/dev/> - шаблонизатор, который является более pythonic чем стандартный джанговский
  - c. <http://docs.python-requests.org/en/latest/> - отличная библиотека для работы с HTTP
11. Далее к изучению пакет, который является набором best practices web разработки WG на питоне: <https://svn.wargaming.net/svn/WOT/web/wgext/trunk/wgext/> (дока: <http://docs.wargaming.net/docs/wgext/en/latest/>). Обязательно знать как минимум состав этого пакета, а желательно и какие-то тонкости реализации. Использование этого пакета значительно ускоряет разработку.
12. Ещё один пакет, который стоит изучить, перед началом разработки web приложений в WG: <https://svn.wargaming.net/svn/WOT/web/wginternal/trunk/wginternal/> (дока: <http://docs.wargaming.net/docs/wginternal/en/latest/>). В поставке с ним идёт работа с OpenID от WGINI, python обвязки над SPA API и другие полезные мелочи.

13. На этом этапе вы уже готовы выполнять задачи на python/django в WG, но для большей уверенности в своих силах рекомендуется пройти 14 пункт, дабы изучить и повзаимодействовать с web проектами компании на практике уже сейчас, а не когда ваш техлид или менеджер решит, что вы готовы.
14. Сделать доработки пункта 9 согласно следующим пунктам:
  - a. Пользователи логинятся не стандартным джанговским способом, а через WGNi, используется wginternal
  - b. Письма пользователям отправляются через WGNc, текст письма "Ах ты %nickname% шалун", используется пакет-клиент для нотифаера
  - c. У каждой категории добавляется признак "Количество боёв у пользователя необходимое для просмотра категории" (может быть 0). Если пользователь пытается просмотреть категорию, но при этом у него недостаточное количество боёв для просмотра, то ему отображается 404, кроме того категория эта не отображается для пользователя на странице категорий.
  - d. Цена товара указывается в голде
  - e. Если у пользователя недостаточно голды (wgwallet), то при нажатии на кнопку оформить заказ ему отображается сообщение "У вас недостаточно голды"
  - f. Если у пользователя достаточно голды, то после нажатия кнопки оформить заказ сначала списываются через wallet голда у пользователя и только при успешном списании отправляется письмо