[Собеседование junior python разработчик Евгений Путрин | Галина Трошкина и Василий Туртугешев - YouTube](https://www.youtube.com/watch?v=LqhbhK8ATLk)

Видео про разбор вопросов на собеседовании по Python:

<https://www.youtube.com/watch?v=BoazgBZ4D7k>

**ДИПЛОМ:**

Серия отличных видео про Django. Всего 26 видео. Очень понятно рассказывает.

В том числе про маршрутизацию, про slug

1.

<https://www.youtube.com/watch?v=FyTL1bnUx5I&list=PLA0M1Bcd0w8xO_39zZll2u1lz_Q-Mwn1F&index=1>

2.

<https://www.youtube.com/watch?v=Ra_fK4X-0eA&list=PLA0M1Bcd0w8xO_39zZll2u1lz_Q-Mwn1F>

3. Маршрутизация, обработка исключений запросов, перенаправления

<https://www.youtube.com/watch?v=af7KvkQORwo&list=PLA0M1Bcd0w8xO_39zZll2u1lz_Q-Mwn1F&index=3>

4. Определение моделей. Миграции: создание и выполнение

https://www.youtube.com/watch?v=k1wZKx6nMjg

5. CRUD - основы ORM по работе с моделями |

https://www.youtube.com/watch?v=-knypkOQSSw

6. Шаблоны. Вывод на Главную страницу Статей.

https://www.youtube.com/watch?v=v3JM4yUFdNo

7. Подключение статических файлов CSS, javascrit, img

https://www.youtube.com/watch?v=IrUG07namQ8

8. Формирование URL-адресов в шаблонах get\_ absolute\_ url

<https://www.youtube.com/watch?v=CFO4aAsUuUk&list=PLA0M1Bcd0w8xO_39zZll2u1lz_Q-Mwn1F&index=8>

9. В самом конце (23 минута) установление связи Категория -> Статьи данной Категории

<https://www.youtube.com/watch?v=tzl1uklqM20&list=PLA0M1Bcd0w8xO_39zZll2u1lz_Q-Mwn1F&index=9>

Сделать у меня также как в этом уроке !!!

10. Начинаем работу с админ-панелью

<https://www.youtube.com/watch?v=Sut1s4LdMG0>

11. Пользовательские теги шаблонов

<https://www.youtube.com/watch?v=Sr-1OBLge6s>

12. Добавляем слаги (slug) к URL-адресам

<https://www.youtube.com/watch?v=Zx6uNIKvVXY>

13. Использование форм, не связанных с моделями

<https://www.youtube.com/watch?v=u37FXeVQIpU>

14. Формы, связанные с моделями. Пользовательские валидаторы

<https://www.youtube.com/watch?v=M6n0Kl_qg3Q>

15. Классы представлений: ListView, DetailView, CreateView

<https://www.youtube.com/watch?v=_0IDdU6PDJo>

16. Основы ORM Django за час

<https://www.youtube.com/watch?v=QrO-YgfWAOU>

17. Mixins - убираем дублирование кода

<https://www.youtube.com/watch?v=5oy0uXpYakg>

18.Постраничная навигация (пагинация)

<https://www.youtube.com/watch?v=pDB9GSlQ7iY>

19. Регистрация пользователей на сайте

<https://www.youtube.com/watch?v=QK4qbVyY7oU>

20. Делаем авторизацию пользователей на сайте

<https://www.youtube.com/watch?v=0jNnU8h2Oz8>

21.Оптимизация сайта с Django Debug Toolbar

https://www.youtube.com/watch?v=qnJ8FJf9RGA&list=PLA0M1Bcd0w8xO\_39zZll2u1lz\_Q-Mwn1F&index=21

22.Включаем кэширование данных

<https://www.youtube.com/watch?v=Z3kry8telRo>

23. Использование капчи captcha

<https://www.youtube.com/watch?v=LL71poFECjE>

24. Тонкая настройка админ панели

https://www.youtube.com/watch?v=6jyl85q6yRs&list=PLA0M1Bcd0w8xO\_39zZll2u1lz\_Q-Mwn1F&index=24

25. Начинаем развертывание Django-сайта на хостинге

https://www.youtube.com/watch?v=reURUKqkyYw

26. Завершаем развертывание Django-сайта на хостинге | Django уроки

https://www.youtube.com/watch?v=reURUKqkyYw

Публикация этого проекта на GitHub:

lesson-10-coolsite.zip: [https://github.com/selfedu-rus/django.](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbHd5aDhUcVZ5dlktTlgtYksyMzBlWFRBdExUd3xBQ3Jtc0tuUEt4RjRKcWZwSHpQMFpGdm40ck91QURWcGRKeTZBTjRUb25naGVYUVRpc3dyVHgtaVEwclpjenZ4czJkVUpMSXRvUms2aEJKeDhpdDJ4X0xjMjFBYXVGYklXZlc0YWdWTHJQazZSdGFSVjQ2dkxQMA&q=https%3A%2F%2Fgithub.com%2Fselfedu-rus%2Fdjango-lessons&v=Sut1s4LdMG0)

Запрос: pycharm сохранить на github

<https://progler.ru/blog/kak-zagruzit-proekt-na-github-iz-pycharm>

Здесь пошаговая инструкция + скриншоты экранов

**Видео1**

cd coolsite

python manage.py runserver

**Видео2**

cd coolsite

python manage.py startapp women - создаем приложение women

**Видео3 (Маршрутизация и обработка исключений)**

cd coolsite

urlpatterns = [  
 path(**''**, index),  
 path(**'cats/<int:catid>/'**, categories),  
]

**def** categories(request, catid):  
 **return** HttpResponse(**f"<h1>Статьи по категориям</><p>{**catid**}</p>"**)

urlpatterns = [  
 path(**''**, index),  
 path(**'cats/<slug:cat>/'**, categories),  
]

**def** categories(request, cat):  
 **return** HttpResponse(**f"<h1>Статьи по категориям</><p>{**cat**}</p>"**)

**Видео4 Создание Моделей. Миграции**

cd coolsite

**from** django.db **import** models  
  
**class** Women (models.Model):  
 titel = models.CharField(max\_length=255)  
 content = models.TextField(blank=**True**)  
 photo = models.ImageField(upload\_to=**"photos/%Y/%m/%d/"**)  
 time\_create = models.DateTimeField(auto\_now\_add=**True**)  
 time\_update = models.DateTimeField(auto\_now=**True**)  
 is\_published = models.BooleanField(default=**True**)

STATIC\_URL = **'static/'**MEDIA\_ROOT = os.path.join(BASE\_DIR, **'media'**)  
MEDIA\_URL = **'/media/'**

**if** settings.DEBUG:  
 urlpatterns += static(settings.MEDIA\_URL, document\_root =

settings.MEDIA\_ROOT)

python -m pip install Pillow

python manage.py makemigrations

- Create model Women

python manage.py migrate

**Видео5 CRUD - основы ORM по работе с моделями (Create, Read, Update, Delete)**

python manage.py shell

Ctrl+Z - выход из shell

>>> from women.models import Women

>>> Women(title = 'Анджелина Джоли', content = 'Биография Анджелины Джоли')

w1 =\_

w1.save()

w2 = Women(title='Энн Хэтэуэй', content='Биография Энн Хэтэуэй')

w3=Women()

w3.title='Джулия Робертс'

w3.content='Биография Джулии Робертс'

w3.save()

**Видео6 Шаблоны. Вывод на Главную страницу Статей. Фильтры**

w3=Women()

**Видео7**

STATIC\_URL = **'static/'**STATIC\_ROOT = os.path.join(BASE\_DIR, **'static'**)  
STATICFILES\_DIRS = []

Нужны данные с GitHub

**Видео8 Формирование URL-адресов в шаблонах**

w3=Women()

**Видео9**

создал суперюзера:

python manage.py createsuperuser

в admin.py зарегистрировал модели

**from** django.contrib **import** admin  
**from** .models **import** Category, Women  
  
*# Register your models here.*admin.site.register(Category)  
admin.site.register(Women)

Наконец-то удалось реализовать алгоритм отбора статей, согласно выбранной категории !!!

Файл проекта: urls.py

**from** django.conf.urls.static **import** static  
**from** django.contrib **import** admin  
**from** django.urls **import** path, include  
  
**from** coolsite **import** settings  
  
urlpatterns = [ *# разделяет ВСЕ поступающие запросы на Два потока:* path(**'admin/'**, admin.site.urls), *# на Админку* path(**''**, include(**"women.urls"**)), *# на urls приложения women*]  
**if** settings.DEBUG:  
 urlpatterns += static(settings.MEDIA\_URL, document\_root =

settings.MEDIA\_ROOT)

Файл приложения: urls.py

**from** django.urls **import** path  
**from** .views **import** \*  
  
urlpatterns = [  
 path(**''**, index, name=**"home"**), *# главному маршруту присвоили имя home* path(**'about/'**, about, name=**"about"**),  
 path(**'post/<int:post\_id>/'**, show\_post, name=**"post"**), *# post\_id прилетает*

*из get\_absolute\_url при открытии страницы show\_post* path(**'category/<int:cat\_id>/'**, show\_category, name=**"category"**)  
]

Фрагмент views.py (по выбранной категории формирует список статей данной категории)

**def** show\_category(request, cat\_id):  
 posts = Women.objects.filter(cat\_id=cat\_id) *# в переменную posts*

*отфильтрованные записи таблицы Women* cats = Category.objects.all()  
 **return** render(request, **'women/index.html'**, {**'cats'**: cats, **'posts'**: posts,

**'menu'**: menu, **'title'**: **'Главная'**})

Файл index.html

{% extends 'women/base.html' %}  
  
{% block content %}  
*<!-- <h1>{{ title }}</h1> -->* <**h3**>Категории</**h3**>  
 {% for c in cats %}  
 <**li**><**a href="{{c.get\_absolute\_url}}"**>{{ c.name }}</**a**></**li**>

*<!- - выводим имя Категории в виде кликабельной ссылки -->* {% endfor %}  
  
 <**br**><**br**>  
 <**h3**>Статьи</**h3**>  
 {% for p in posts %}  
 <**h4**>{{ p.title }}</**h4**> *<!-- выводим имя Женщины -->* {% endfor %}  
{% endblock %}

Файл models.py

**from** django.db **import** models  
**from** django.urls **import** reverse  
  
**class** Women (models.Model):  
 title = models.CharField(max\_length=255)  
 content = models.TextField(blank=**True**)  
 photo = models.ImageField(upload\_to=**"photos/%Y/%m/%d/"**)  
 time\_create = models.DateTimeField(auto\_now\_add=**True**)  
 time\_update = models.DateTimeField(auto\_now=**True**)  
 is\_published = models.BooleanField(default=**True**)  
 cat = models.ForeignKey(**"Category"**, on\_delete=models.PROTECT, null=**True**)  
  
 **def** \_\_str\_\_(self):  
 **return** self.title  
  
 **def** get\_absolute\_url(self):  
 **return** reverse (**'post'**, kwargs = {**'post\_id'**: self.pk})  
 *# возвращает значение переменной post\_id,*

*равной значению pk (primary key) статьи***class** Category (models.Model):  
 name = models.CharField(max\_length=100, db\_index=**True**)  
  
 **def** \_\_str\_\_(self):  
 **return** self.name  
  
 **def** get\_absolute\_url(self):  
 **return** reverse (**'category'**, kwargs = {**'cat\_id'**: self.pk})

*# возвращает значение переменной cat\_id,*

*равной значению pk (primary key) категории*

**ПОВТОРЯЮ ОДИН В ОДИН ПРОЕКТ ПРЕПОДАВАТЕЛЯ blog**

1.Открыл PyCharm, прерывая загрузку New Project

2.Новый Проект: D:\Python(Диплом)\diplom

Галочек нигде нет Create => PyCharm сам создал необходимые папки

3.Установил окружение Setting….. => появилась папка venv

4.В Терминале проверил: python –V pip -V

5.В PyCharm установил Django: django install => в окружении venv появился Django

6.В Терминале создал проект: django-admin startproject blog

7.Перешел в папку blog: cd blog - чтобы был доступен файл manage.py

(чтобы подняться вверх: cd.. посмотреть содержимое папки: dir)

8.Запустил Сервер: python manage.py runserver => появилась страница Django

9.В папке настроек (settings.py) установил русский язык: LANGUAGE\_COD: “ru-ru”

10. python manage.py startapp blogapp - создал приложение blogapp

11.В файле приложения models.py создал модели: Category, Product

12.В файле settings.py подключил приложение

INSTALLED\_APPS = [

………………..

‘blogapp’

]

13.Сделал миграции, чтобы произошли изменения в БД

python manage.py makemigrations - создать миграции

- Create model Category

- Create model Product

python manage.py migrate - выполнить миграции

14.Создал суперпользователя: python manage.py createsuperuser ivan 12345

15.Подключил к Админке наши Модели в файле admin.py

**from** django.contrib **import** admin  
**from** .models **import** Category,Product  
  
*# Register your models here.*admin.site.register(Category)  
admin.site.register(Product)

16.Зашел в Админку и ввел данные:

Вязаные

Зайка 1500

Мишка 1400

Канзаши

Брошь 1300

Цветок 1200

17.В папке приложения blogapp создал Пакет Python Package management

(чтобы через него запускать свои скрипты)

18.В папке management создаем пакет commands

19.В папке commands создал файл fill\_db.py

**from** django.core.management.base **import** BaseCommand  
  
**class** Command(BaseCommand):  
 **def** handle(self, \*args, \*\*options):  
 print(**"Это моя команда"**)

20.В Терминале выполнил команду: python manage.py fill\_db

=> выполнится команда print(**"Это моя команда"**)

21.Потренировался выполнять разные запросы:

*# Выбираем ВСЕ категории  
# categories = Category.objects.all()  
# print(categories)  
# for item in categories:  
# print(item) # печатаем объект  
# print(item.name) # печатаем поле name  
# print(type(item))  
# print("End")  
  
# Выбираем ОДНУ категорию  
# categori = Category.objects.get(name="Игрушки")  
# print(categori)  
  
# Выбираем несколько спомощью фильтра:  
# categori = Category.objects.filter(name="Игрушки")  
# print(categori)  
  
# categori = Post.objects.filter(name ="Танк")  
# print(categori)*

**from** django.core.management.base **import** BaseCommand  
**from** blogapp.models **import** Category, Product *#не подкрашивать самый верхний*

*# blog правой мышкой Mark Sourse Root***class** Command(BaseCommand):  
 **def** handle(self, \*args, \*\*options):  
 *#print("Это моя команда")  
 # Выбираем товары, принадлежащие категории "Канзаши" !!!* products = Product.objects.filter(category\_\_name=**"Канзаши"**)  
 print(products)  
 **for** el **in** products: *# пробегаем в цикле по выбранным товарам* print(el.name) *# для каждого товара печатаем значение поля name*

*#Product.objects.filter(company\_\_name="Apple")*

**ВНИМАНИЕ:** в category\_\_name внизу ДВА подчеркивания !!!

Это нашел на **METANIT.COM** : <https://metanit.com/python/django/5.3.php>

--------------------------------------------------------------------------------------

21.В самом верхнем blog создал папку templates и скопировал туда файлы **karma**

22.В файле views.html

23.В папке blogapp создаем папку templates

24. В папке templates создаем папку blogapp

25.В эту папку скопировал файл Главной страницы index.html

26.В папке приложения blogapp создал еще один файл адресов: urls.py

27.В этом файле написал адреса:

28.Запустил Сервер командой: python manage.py runserver

=> открылась Главная страница index.html. Но без стилей CSS. Надо подключить СТАТИКУ

ПОДКЛЮЧАЕМ СТАТИКУ

29.В самом верхнем blog создал папку static

30.В эту папку копируем папки статики и файлы js

31.В файле settings.py дописываем путь где искать файлы СТАТИКИ

STATIC\_URL = **'static/'** STATICFILES\_DIRS = [os.path.join(BASE\_DIR, **'static'**)]

32.В index.html для тегов <**link rel="/static/stylesheet"**  добавил /static/

И для подключения файлов js <**script src="/static/js/vendor/**

33.Запустил Сервер командой: python manage.py runserver

Для очистки кэша в браузере Microsoft Edge нажмите клавиши Ctrl + Shift + Del на клавиатуре или:

В правом верхнем углу браузера нажмите на значок в виде трех точек

Выберите Журнал > Дополнительные параметры > Удалить данные о просмотре веб-страниц

В открывшемся окне, выберите временной диапазон «Все время»

Отметьте галочками пункты Кэшированные изображения и файлы и Файлы cookie и друге данные сайтов. Остальные галочки можно снять

Нажмите кнопку Удалить сейчас

=> теперь страница файла КРАСИВАЯ - с разметкой !!!