

Django

<https://www.djangoproject.com/>

Веб-фреймворк, быстрая разработка,
удобные хелперы, почти всё что нужно в
коробке. Чего в коробке нет - куча
дополнений

<https://goo.gl/gHF3Wy>

Джанго Рейнхардт



Но я же не знаю Python...

http://pythontutor.ru/lessons/inout_and_arithmetic_operations/

<https://tvorog.me/basics.html>

Что мы будем делать

<https://docs.djangoproject.com/en/2.0/intro/tutorial01/>

Подведение итогов за день - выполнение всех тестов в этом репозитории

<https://github.com/Kulikovpavel/goto201807django>

Виртуальное окружение

`pip install` глобально

глобально

`pip install -- user`

для юзера

`python -m venv /path/to/new/venv`

virtual environment

`source /path/to/new/venv/bin/activate`

`/path/to/new/venv/Scripts/activate`

Но есть способ лучше

`pipenv!`

Pipenv

<https://docs.pipenv.org/>

```
pipenv --python 3.6
```

```
pipenv install --dev
```

```
pipenv run pip freeze
```

```
pipenv shell
```

```
pipenv --where, pipenv --venv, pipenv --py
```

Наконец то django

```
pipenv --python 3.6
```

```
pipenv install django
```

```
pipenv shell
```

```
cd project_dir
```

```
django-admin startproject mytestproject .
```

```
python manage.py startapp main
```

```
python manage.py runserver
```

Первые шаги, View

```
main/views.py
```

```
from django.http import HttpResponse
```

```
def index(request):  
    return HttpResponse("Hello, world. You're at the main index.")
```

urls

main/urls.py

```
from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index'),

]
```

mytestproject/urls.py

```
from django.contrib import admin
from django.urls import include, path

urlpatterns = [
    path('main/', include('main.urls')),
    path('admin/', admin.site.urls),
]
```


Данные, модели!

```
main/models.py
```

```
from django.db import models
```

```
class Question(models.Model):  
    question_text = models.CharField(max_length=200)  
    pub_date = models.DateTimeField('date published')
```

```
class Choice(models.Model):  
    question = models.ForeignKey(Question, on_delete=models.CASCADE)  
    choice_text = models.CharField(max_length=200)  
    votes = models.IntegerField(default=0)
```

Подробнее

- [AutoField](#)
- [BigAutoField](#)
- [BigIntegerField](#)
- [BinaryField](#)
- [BooleanField](#)
- [CharField](#)
- [DateField](#)
- [DateTimeField](#)
- [DecimalField](#)
- [DurationField](#)
- [EmailField](#)
- [FileField](#)
 - [FileField and FieldFile](#)
- [FilePathField](#)
- [FloatField](#)
- [ImageField](#)
- [IntegerField](#)
- [GenericIPAddressField](#)
- [NullBooleanField](#)

- [PositiveIntegerField](#)
- [PositiveSmallIntegerField](#)
- [SlugField](#)
- [SmallIntegerField](#)
- [TextField](#)
- [TimeField](#)
- [URLField](#)
- [UUIDField](#)

- [Relationship fields](#)
 - [ForeignKey](#)
 1. [Database Representation](#)
 2. [Arguments](#)
 - [ManyToManyField](#)
 1. [Database Representation](#)
 2. [Arguments](#)
 - [OneToOneField](#)
-

Накатываем на базу данных

```
mysite/settings.py
```

```
INSTALLED_APPS = [  
    'main.apps.MainConfig',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

```
$ python manage.py makemigrations main
```

```
// смотрим на то что получилось, опционально
```

```
$ python manage.py sqlmigrate main 0001
```

```
$ python manage.py migrate
```

Дальнейшие изменения моделей

- 1) Делаем изменения
- 2) `python manage.py makemigrations`
- 3) `python manage.py migrate`

ORM - object-relation mapping

<https://docs.djangoproject.com/en/2.0/topics/db/queries/>

```
$ python manage.py shell
```

и вперед

```
from django.db import models

class Question(models.Model):
    # ...
    def __str__(self):
        return self.question_text

class Choice(models.Model):
    # ...
    def __str__(self):
        return self.choice_text
```

Админка

```
python manage.py createsuperuser
```

```
main/admin.py
```

```
from django.contrib import admin
```

```
from .models import Question
```

```
admin.site.register(Question)
```

Еще о view

main/views.py

```
def detail(request, question_id):  
    return HttpResponse("You're looking at question %s."  
% question_id)
```

```
def results(request, question_id):  
    response = "You're looking at the results of  
question %s."  
    return HttpResponse(response % question_id)
```

```
def vote(request, question_id):  
    return HttpResponse("You're voting on question %s."  
% question_id)
```

main/urls.py

```
from django.urls import path
```

```
from . import views
```

```
urlpatterns = [  
    # ex: /main/  
    path('', views.index, name='index'),  
    # ex: /main/5/  
    path('<int:question_id>/', views.detail,  
name='detail'),  
    # ex: /main/5/results/  
    path('<int:question_id>/results/', views.results,  
name='results'),  
    # ex: /main/5/vote/  
    path('<int:question_id>/vote/', views.vote,  
name='vote'),  
]
```


Шаблоны!

main/templates/main/index.html

```
{% if latest_question_list %}
    <ul>
        {% for question in latest_question_list %}
            <li><a href="/main/{{ question.id }}">{{
question.question_text }}</a></li>
        {% endfor %}
    </ul>
{% else %}
    <p>No main are available.</p>
{% endif %}
```

main/views.py

```
from django.shortcuts import render

from .models import Question

def index(request):
    latest_question_list =
Question.objects.order_by('-pub_date')[:5]
    context = {'latest_question_list':
latest_question_list}
    return render(request, 'main/index.html',
context)
```

А что если...

main/views.py

```
from django.shortcuts import get_object_or_404, render
```

```
from .models import Question
```

```
# ...
```

```
def detail(request, question_id):
```

```
    question = get_object_or_404(Question, pk=question_id)
```

```
    return render(request, 'main/detail.html', {'question': question})
```

Плохо - `{{ question.question_text }}`

Хорошо - `{{ question.question_text }}`

...

the 'name' value as called by the {% url %} template tag
`path('<int:question_id>/', views.detail, name='detail'),`

...

Forms

main/templates/main/detail.html

```
<h1>{{ question.question_text }}</h1>
```

```
{% if error_message %}<p><strong>{{ error_message }}</strong></p>{% endif %}
```

```
<form action="{% url 'vote' question.id %}" method="post">
```

```
{% csrf_token %}
```

```
{% for choice in question.choice_set.all %}
```

```
    <input type="radio" name="choice" id="choice{{ forloop.counter }}" value="{{ choice.id }}" />
```

```
    <label for="choice{{ forloop.counter }}">{{ choice.choice_text }}</label><br />
```

```
{% endfor %}
```

```
<input type="submit" value="Vote" />
```

```
</form>
```

main/urls.py

```
path('<int:question_id>/vote/', views.vote, name='vote'),
```

main/views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import get_object_or_404, render
from django.urls import reverse

from .models import Choice, Question
# ...
def vote(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        # Redisplay the question voting form.
        return render(request, 'main/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        # Always return an HttpResponseRedirect after successfully dealing
        # with POST data. This prevents data from being posted twice if a
        # user hits the Back button.
        return HttpResponseRedirect(reverse('results', args=(question.id,)))
```

Generics, for self-study

<https://docs.djangoproject.com/en/2.0/topics/class-based-views/>

Статика

```
main/static/main/style.css
```

```
li a {  
    color: green;  
}
```

```
main/templates/main/index.html
```

```
{% load static %}
```

```
<link rel="stylesheet" type="text/css" href="{% static 'main/style.css' %}" />
```

```
main/static/main/style.css
```

```
body {  
    background: white url("images/background.gif") no-repeat;  
}
```

Деплой

<https://docs.djangoproject.com/en/2.0/howto/static-files/deployment/>

Об этом вам расскажет Филипп завтра

Вопросы?

Павел Куликов

kulikovpavel@gmail.com

+79031183741, tg, whatsapp