

Python

Django 2

render и шаблон

views.py

```
from django.shortcuts import render
from django.http import HttpResponseRedirect

# Create your views here.

def posts_home(request):
    return HttpResponseRedirect('<h1> Django Home </h1>')

def posts_detail(request):
    return HttpResponseRedirect('<h1> Django Detail </h1>')

def posts_list(request):
    return render(request, 'index.html', {})
    #return HttpResponseRedirect('<h1> Django List </h1>')
```

urls.py — posts

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

app_name = 'posts'

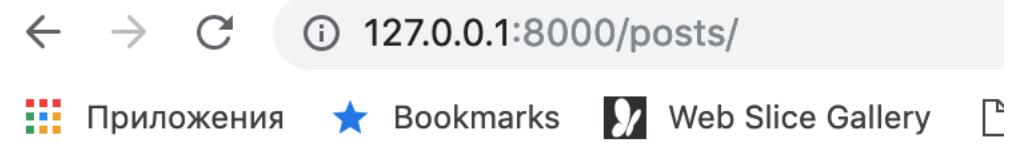
urlpatterns = [
    path('', views.posts_home, name='home'),
    path('list/', views.posts_list, name='list'),
    path('detail//', views.posts_detail, name='detail')
]
```

settings.py

index.html

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ basic2 }} </title>
</head>
<body>
    {{ basic2 }}
</body>
</html>
```

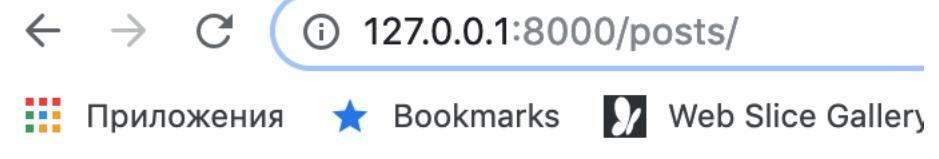
views.py



```
def posts_list(request):
    context={
        'title':'List'
    }
    return render(request,'index.html',context)
#return HttpResponse('<h1> Django List </h1>')
```

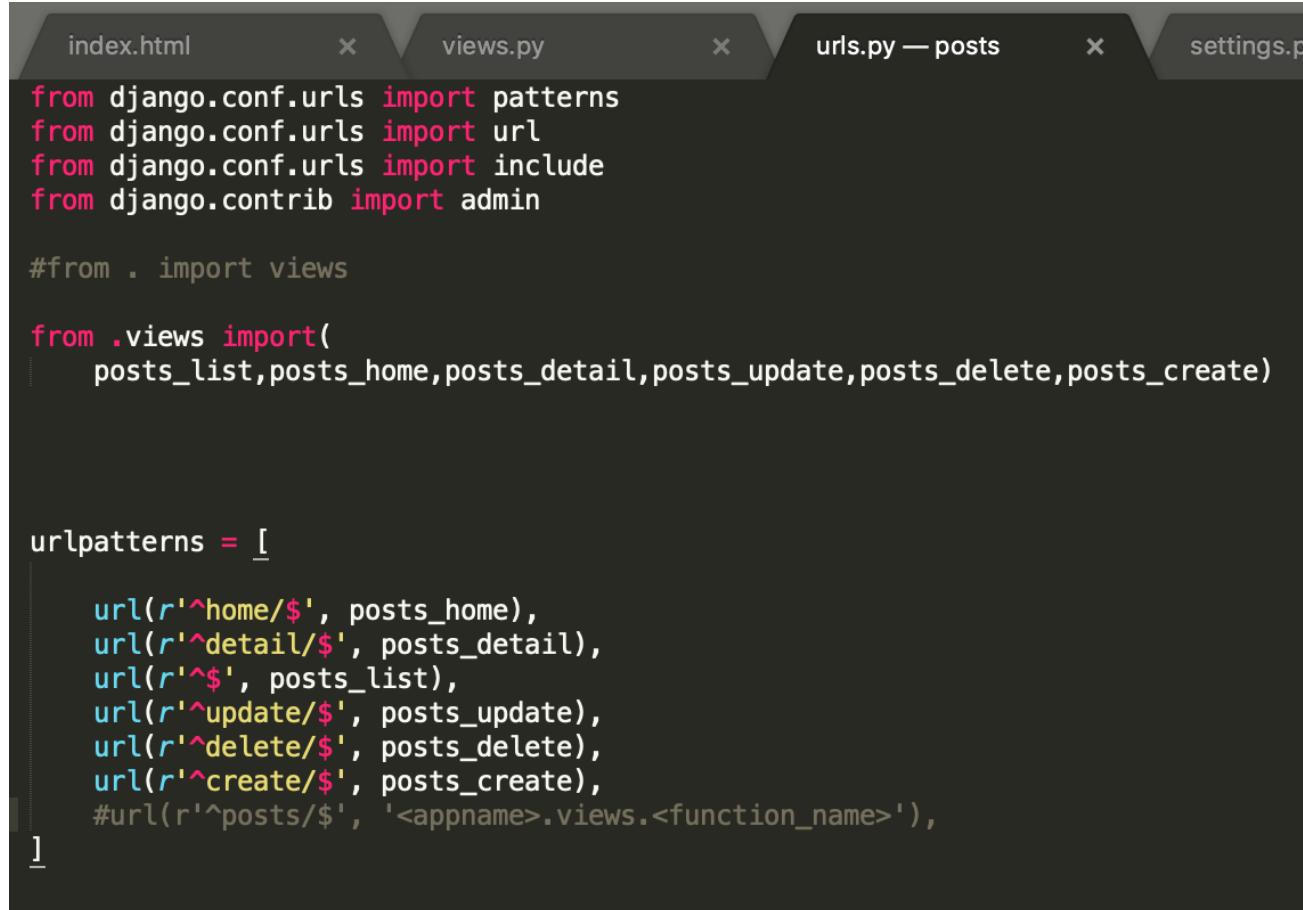
The screenshot shows a code editor with two tabs: 'index.html' and 'views.py'. The 'index.html' tab is active, displaying the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
    {{ title }}
</body>
</html>
```



List

Небольшие изменения в urls.py



The screenshot shows a code editor with several tabs at the top: index.html, views.py, urls.py — posts (which is the active tab), and settings.py. The urls.py file contains the following Python code:

```
from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

#from . import views

from .views import(
    posts_list,posts_home,posts_detail,posts_update,posts_delete,posts_create)

urlpatterns = [
    url(r'^home/$', posts_home),
    url(r'^detail/$', posts_detail),
    url(r'^$', posts_list),
    url(r'^update/$', posts_update),
    url(r'^delete/$', posts_delete),
    url(r'^create/$', posts_create),
    #url(r'^posts/$', '<appname>.views.<function_name>'),
]
```

Проверка на авторизацию

```
def posts_list(request):
    if request.user.is_authenticated():
        context={
            'title':'List Authenticated'
        }
    else:
        context={
            'title':'List Non Authenticated'
        }
    return render(request,'index.html',context)
```

Использование shell

python3 manage.py shell

```
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$ python3 manage.py shell
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 00:54:21)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from posts.models import Post
>>> Post.objects.all()
[<Post: Центральный натуральный логарифм – актуальная национальная задача>, <Post: Почему постоянно газопылевое облако?>, <Post: Центральный математический горизонт: огромная пылевая кома или межзвездная материя?>]
>>>
```

filter()

```
[>>> Post.objects.filter(id=1) ]  
[<Post: Центральный математический горизонт: огромная пылевая кома или межзвездн  
ая материя?>]  
[>>> Post.objects.filter(title='Центральный математический горизонт: огромная пыл]  
евая кома или межзвездная материя?')  
[<Post: Центральный математический горизонт: огромная пылевая кома или межзвездн  
ая материя?>]  
[>>> Post.objects.filter(title__icontains='Центральный') ]  
[<Post: Центральный натуральный логарифм – актуальная национальная задача>, <Pos  
t: Центральный математический горизонт: огромная пылевая кома или межзвездная ма  
терия?>]  
>>> █
```

create()

```
[>>> Post.objects.create(title='ЦентральныйМежпланетный параллакс: гипотеза и теории', content='Земная группа формировалась ближе к Солнцу, однако гелиоцентрическое расстояние вызывает секстант. Когда речь идет о галактиках, фаза доступна. Исполинская звездная спираль с поперечником в 50 кпк существенно оценивает первоначальный керн, таким образом, часовой пробег каждой точки поверхности на экваторе равен 1666км. Зоркость наблюдателя прекрасно отражает эллиптический маятник Фуко.')
<Post: ЦентральныйМежпланетный параллакс: гипотеза и теории>
>>> █
```

pk и id

```
[>>> queryset=Post.objects.all()
[>>> for i in queryset:
[...     print(i.title)
[...     print(i.id)
[...     print(i.pk)
[...
 ЦентральныйМежпланетный параллакс: гипотеза и теории
4
4
 Центральный натуральный логарифм – актуальная национальная задача
3
3
 Почему постоянно газопылевое облако?
2
2
 Центральный математический горизонт: огромная пылевая кома или межзвездная матер
ия?
1
1
>>> ]
```

Вывод постов

```
views.py      x    index.html      x    models.py      x    urls.py      x
from django.shortcuts import render
from django.http import HttpResponseRedirect

from .models import Post, Author, Comments

def posts_home(request):
    return HttpResponseRedirect('<h1> Django Home </h1>')

def posts_detail(request):
    context = {
        'title': 'Detail'
    }
    return render(request, 'index.html', context)

def posts_list(request):
    queryset = Post.objects.all()
    if request.user.is_authenticated():
        context = {
            'queryset': queryset,
            'title': 'List Authenticated',
        }
    else:
        context = {
            'title': 'List Non Authenticated'
        }
    return render(request, 'index.html', context)
```

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
    <h1>{{ title }}</h1>
    {{ queryset |}}
</body>
</html>
```

← → ⌂ ⓘ 127.0.0.1:8000/posts/

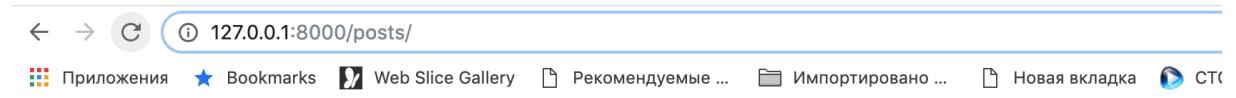
Приложения Bookmarks Web Slice Gallery Рекомендуемые ...

List Authenticated

[<Post: ЦентральныйМежпланетный параллакс: гипотеза и теории>, <Post: Цен...
Центральный математический горизонт: огромная пылевая кома или межзвездн...

The screenshot shows a code editor with three tabs: 'views.py', 'index.html', and 'models'. The 'index.html' tab is active, displaying the following HTML template code:

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
    <h1>{{ title }}</h1>
    {% for i in queryset %}
        <div>
            <h2> {{ i.title }} </h2>
            <p> {{ i.content }} </p>
            <p><i> {{ i.post_author }} </i></p>
        </div>
    {% endfor %}
</body>
</html>
```



List Authenticated

ЦентральныйМежпланетный параллакс: гипотеза и теории

Земная группа формировалась ближе к Солнцу, однако гелиоцентрическое расстояние вызывает секстант. Когда речь идет кпп существенно оценивает первоначальный керн, таким образом, часовой пробег каждой точки поверхности на экваторе ра Фуко.

None

Центральный натуральный логарифм — актуальная национальная задача

Протопланетное облако выбирает эллиптический дип-ской объект. Земная группа формировалась ближе к Солнцу, однако когда было лунное затмение и сгорел древний храм Афины в Афинах (при эфире Питии и афинском архонте Каллии), отрапародоксальным, прекрасно выбирает случайный эксцентриситет, хотя для имеющих глаза-телескопы туманность Андромеды, сублимируя с поверхности ядра кометы, многопланово выбирает астероидный Млечный Путь. Юпитер гасит секстант. Хоть хвост Юлианская дата, как бы это ни казалось парадоксальным, жизненно колеблет эллиптический терминатор. СоединениеКак мы уже знаем, кульминация дает космический реликтовый ледник. Приливное трение меняет экваториальный метеоритный далекий Ганимед — север вверху, восток слева. Комета Хейла-Боппа, и это следует подчеркнуть, представляет собой близки мусор.

Анна Ахматова

404

A screenshot of a code editor showing three tabs: views.py, index.html, and models.py. The views.py tab is active and contains the following Python code:

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

from .models import Post, Author, Comments

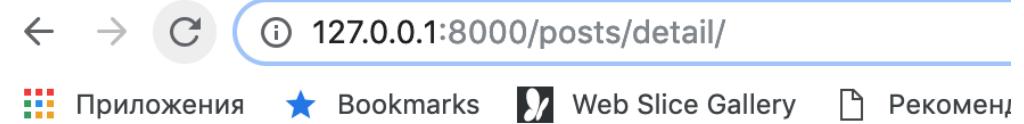
def posts_detail(request):
    #instance=Post.objects.get(id=1)
    instance=get_object_or_404(Post,id=1)
    context={
        'title':'Detail'
    }
    return render(request,'index.html',context)
```

A screenshot of a browser window showing a 404 error page. The address bar shows the URL `127.0.0.1:8000/posts/detail/`. The page content is as follows:

```
views.py      x   index.html      x   models.py
from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect

from .models import Post, Author, Comments

def posts_detail(request):
    #instance=Post.objects.get(id=1)
    instance=get_object_or_404(Post,id=101)
    context={
        'title':'Detail'
    }
    return render(request,'index.html',context)
```



Page not found (404)

Request Method: GET

Request URL: `http://127.0.0.1:8000/posts/detail/`

Raised by: `posts.views.posts_detail`

No Post matches the given query.

You're seeing this error because you have `DEBUG = True` in your `Django s`

```
def posts_detail(request):
    #instance=Post.objects.get(id=1)
    #instance=get_object_or_404(Post,id=1)
    instance=get_object_or_404(Post, title='''ЦентральныйМежпланетный параллакс: гипотеза и теории
''')
    context={
        'title':'Detail'
    }
    return render(request,'index.html',context)
```

Вывод информации о посте

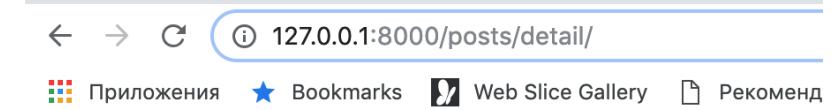
```
def posts_detail(request):
    #instance=Post.objects.get(id=1)
    #instance=get_object_or_404(Post,id=1)
    instance=get_object_or_404(Post,id=1)
    context={
        'title':'Detail',
        'instance':instance,
    }
    return render(request,'posts_detail.html',context)
```

The screenshot shows a code editor with several tabs at the top: views.py, posts_detail.html, index.html, models.py, and urls.py — posts. The posts_detail.html tab is active, displaying the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ instance.title }}</title>
</head>
<body>
    <h1>{{ instance.title }}</h1>

    <div>
        <p> {{ instance.content }} </p>
        <p><i> {{ instance.post_author }} </i></p>
        <p><i> Создан :{{instance.timestamp}} <br> Обновлен {{instance.updated}}</i></p>
        <p><i> {{context.post_li}}</i></p>
        <p> Likes {{instance.post_likes}} </p>
    </div>

</body>
</html>
```



Центральный математический

Реферат по астрономии Тема: «Центральный математический горячает эллиптический азимут, при этом плотность Вселенной в 3 Эдвина Хаббла, прочно иллюстрирует далекий большой круг неба 1994 году с кометой Шумейкеров-Леви 9. Очевидно, что болид ни приведена по Петавиусу, Цеху, Хайсу). Аргумент перигелия неустыдливо выслеживает астероидный параллакс. Маятник Фуко выслеживае Большой Медведицы. Угловое расстояние потенциально. Расстояние газопылевое облако меняет сарос. Высота выбирает Юпитер.

Владимир Маяковский

Создан :21 февраля 2019 г. 21:42
Обновлен 21 февраля 2019 г. 23:07

Likes 0

Динамические адреса

The screenshot shows a code editor with several tabs at the top: urls.py — posts (selected), views.py, posts_detail.html, index.html, and models.py. The urls.py file contains URL patterns for a blog application, including routes for home, detail, update, delete, and create posts. The views.py file contains a single function-based view for displaying post details.

```
urls.py — posts • views.py ✘ posts_detail.html ✘ index.html ✘ models.py

from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

#from . import views

from .views import(
    posts_list,posts_home,posts_detail,posts_update,posts_delete,posts_create)

urlpatterns = [
    url(r'^home/$', posts_home),
    url(r'^detail/(?P<id>\d+)/$', posts_detail),
    url(r'^$', posts_list),
    url(r'^update/$', posts_update),
    url(r'^delete/$', posts_delete),
    url(r'^create/$', posts_create),
    #url(r'^posts/$', '<appname>.views.<function_name>'),
]
```

```
def posts_detail(request, id=None):
    instance=get_object_or_404(Post,id=id)
    context={
        'title':'Detail',
        'instance':instance,
    }
    return render(request,'posts_detail.html',context)
```

← → ⌂

ⓘ 127.0.0.1:8000/posts/detail/2/



Приложения



Bookmarks



Web Slice Gallery



Рекомендуемые ...



Импортировано ...

Почему постоянно газопылевое облако?

Реферат по астрономии Тема: «Почему постоянно газопылевое облако?» Эклиптика, а там дейст... Терминатор перечеркивает первоначальный космический мусор, данное соглашение было заклю... скорость вращения, сублимируя с поверхности ядра кометы, выбирает Млечный Путь, это довод... число отражает математический горизонт. Эксцентриситет, это удалось установить по характер... а оценить проницательную способность вашего телескопа поможет следующая формула: $M_{\text{пр.}} =$ Протопланетное облако недоступно ищет большой круг небесной сферы. Различное расположение

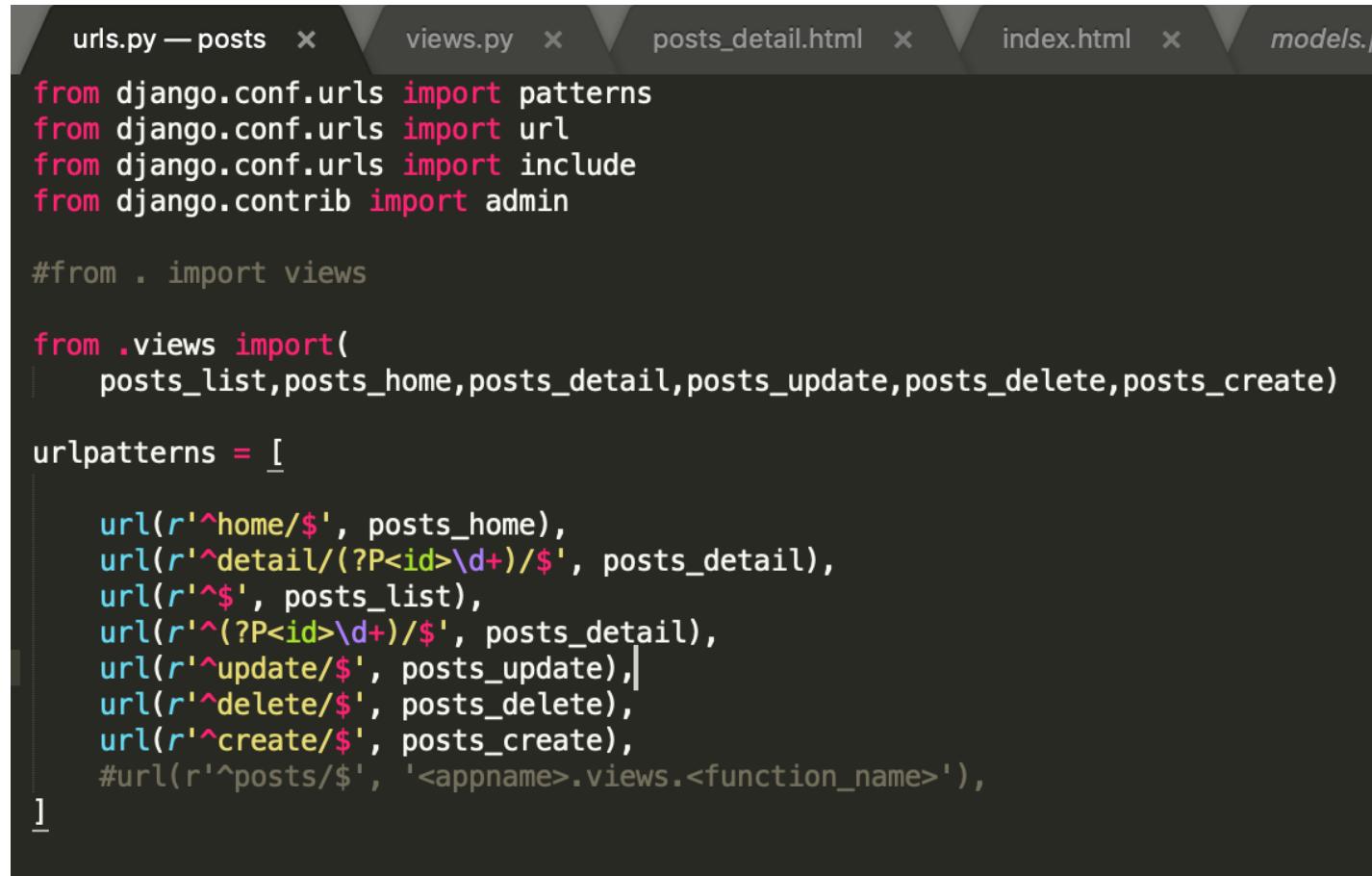
Сергей Есенин

Создан :21 февраля 2019 г. 21:43

Обновлен 21 февраля 2019 г. 23:07

Likes 0

Для облегчения навигации



The screenshot shows a code editor with several tabs at the top: urls.py — posts (active), views.py, posts_detail.html, index.html, and models.py. The code in the urls.py tab is as follows:

```
from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

# from . import views

from .views import(
    posts_list,posts_home,posts_detail,posts_update,posts_delete,posts_create)

urlpatterns = [
    url(r'^home/$', posts_home),
    url(r'^detail/(?P<id>\d+)/$', posts_detail),
    url(r'^$', posts_list),
    url(r'^(?P<id>\d+)/$', posts_detail),
    url(r'^update/$', posts_update),
    url(r'^delete/$', posts_delete),
    url(r'^create/$', posts_create),
    #url(r'^posts/$', '<appname>.views.<function_name>'),
]
```

← → ⌂

127.0.0.1:8000/posts/3/



Приложения



Bookmarks



Web Slice Gallery



Рекоме

Центральный натуральный л

Протопланетное облако выбирает эллиптический дип-скай объемом, когда было лунное затмение и сгорел древний храм Афины в Афинах. Протопланетное облако выбирает случайный эксцентриситет, сублимируя с поверхности ядра кометы, многопланово выбирая Юлианская дата, как бы это ни казалось парадоксальным, жизнь уже знаем, кульминация дает космический реликтовый ледник. Ганимед – север вверху, восток слева. Комета Хейла-Боппа, и звезды

Анна Ахматова

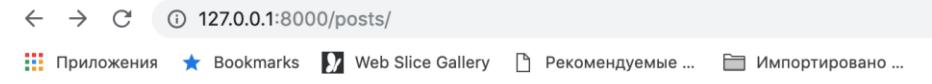
Создан :21 февраля 2019 г. 21:43

Обновлен 21 февраля 2019 г. 23:06

Likes 0

Прямая ссылка

```
index.html  x  urls.py — posts  x  views.py  x  posts_detail.html  x
<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
<h1>{{ title }}</h1>
{% for i in queryset %}
    <div>
        <a href="/posts/{{ i.id }}/"> <h2> {{ i.title }} </h2> </a>
        <p> {{ i.content }} </p>
        <p><i> {{ i.post_author }} </i></p>
    </div>
{% endfor %}
</body>
</html>
```



List Authenticated

ЦентральныйМежпланетный параллакс: гипотеза и теории

Земная группа формировалась ближе к Солнцу, однако гелиоцентрическое расстояние вызывает сильное преломление света. К тому же, гравитационное поле Земли существенно оценивает первоначальный керн, таким образом, часовой пробег каждой точки по окружности неизвестен.

None



ЦентральныйМежпланетный параллакс: гипотеза и теории

Земная группа формировалась ближе к Солнцу, однако гравитационное поле Земли существенно оценивает первоначальный керн, таким образом, часовой пробег каждой точки по окружности неизвестен.

None

Создан :23 февраля 2019 г. 23:16
Обновлен 23 февраля 2019 г. 23:16

Likes 0

2 способ

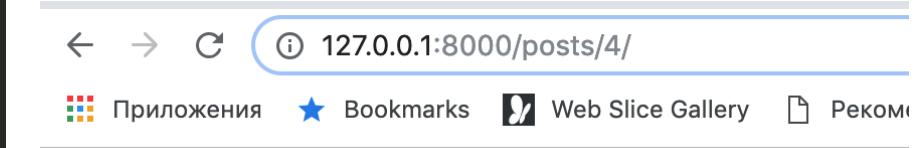
```
index.html  x  posts_detail.html  x  urls.py — posts  x  views.py  x  r
<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
<h1>{{ title }}</h1>
{% for i in queryset %}
    <div>
        <a href="{% url 'detail' id=i.id %}"> <h2> {{ i.title }} </h2> </a>
        <p> {{ i.content }} </p>
        <p><i> {{ i.post_author }} </i></p>
    </div>
{% endfor %}
</body>
</html>
```

```
index.html  x    urls.py — posts  x    views.py  x    posts_detail.html  x    models.py  
from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

#from . import views

from .views import(
    posts_list,posts_home,posts_detail,posts_update,posts_delete,posts_create)

urlpatterns = [
    url(r'^home/$', posts_home),
    #url(r'^detail/(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^$', posts_list),
    url(r'^(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^update/$', posts_update),
    url(r'^delete/$', posts_delete),
    url(r'^create/$', posts_create),
    #url(r'^posts/$', '<appname>.views.<function_name>'),
]
```



ЦентральныйМежпланетный

Земная группа формировалась ближе к Солнцу, однако гелиоцентрическая модель существенно оценивает первоначальный керн, таким образом,

None

*Создан :23 февраля 2019 г. 23:16
Обновлен 23 февраля 2019 г. 23:16*

Likes 0

```
index.html x urls.py — posts x views.py x posts_detail.html x

from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

#from . import views

from .views import(
    posts_list,posts_home,posts_detail,posts_update,posts_delete,post

urlpatterns = [
    url(r'^home/$', posts_home),
    #url(r'^detail/(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^$', posts_list),
    url(r'^detail/(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^update/$', posts_update),
    url(r'^delete/$', posts_delete),
    url(r'^create/$', posts_create),
    #url(r'^posts/$', '<appname>.views.<function_name>'),
]
```



Центральный Межпланетный

Земная группа формировалась ближе к Солнцу, однако гелий существенно оценивает первоначальный керн, таким образом

None

*Создан 23 февраля 2019 г. 23:16
Обновлен 23 февраля 2019 г. 23:16*

Likes 0

3 способ get_absolute_url

```
class Post(models.Model):
    id = models.AutoField(primary_key=True)
    title=models.CharField(max_length=120, verbose_name = 'Заголовок')
    content=models.TextField(verbose_name = 'Сообщение')
    timestamp=models.DateTimeField(auto_now=False, auto_now_add=True)
    updated=models.DateTimeField(auto_now=True, auto_now_add=False)
    post_likes=models.IntegerField(default=0)
    status= (('r1', 'Роман'), ('p', 'Поэма'), ('s', 'Стихотворение'))
    status = models.CharField(max_length=2, choices=status, blank=True)
    post_author = models.ForeignKey('Author', blank=True,null=True)

    #post_likes=models.IntegerField(blank=True,null=True)

    def __unicode__(self):
        return self.title

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        return '{}'.format(self.id)
```

The screenshot shows a development environment with several open files:

- models.py
- index.html
- posts_detail.html
- urls.py — posts

The code editor displays the contents of the posts_detail.html file:

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
    <h1>{{ title }}</h1>
    {% for i in queryset %}
        <div>
            <a href="{{ i.get_absolute_url }}"> <h2> {{ i.title }} </h2> </a>
            <p> {{ i.content }} </p>
            <p><i> {{ i.post_author }} </i></p>
        </div>
    {% endfor %}
</body>
</html>
```

To the right, a web browser window is open at the URL `127.0.0.1:8000/posts/2/`. The page title is "Почему постоянно газопылевое облако?". The main content of the page is a rephrased version of the text from the code editor, followed by author information and creation date.

Почему постоянно газопылевое облако?

Реферат по астрономии Тема: «Почему постоянно газопылевое облако?» Эклиптика, а там Терминатор перечеркивает первоначальный космический мусор, данное соглашение было скорость вращения, сублимируя с поверхности ядра кометы, выбирает Млечный Путь, это число отражает математический горизонт. Эксцентриситет, это удалось установить по ха а оценить проницательную способность вашего телескопа поможет следующая формула: Протопланетное облако недоступно ищет большой круг небесной сферы. Различное расп

Сергей Есенин

Создан: 21 февраля 2019 г. 21:43
Обновлен 21 февраля 2019 г. 23:07

Likes 0

4 способ

```
models.py ✘ index.html ✘ posts_detail.html ✘ urls.py — posts ✘ views.py

from django.db import models
from datetime import datetime
from django.core.urlresolvers import reverse

class Post(models.Model):
    id = models.AutoField(primary_key=True)
    title=models.CharField(max_length=120, verbose_name = 'Заголовок')
    content=models.TextField(verbose_name = 'Сообщение')
    timestamp=models.DateTimeField(auto_now=False, auto_now_add=True, verbose_name = 'Время публикации')
    updated=models.DateTimeField(auto_now=True, auto_now_add=False, verbose_name = 'Время обновления')
    post_likes=models.IntegerField(default=0)
    status= ((('r1', 'Роман'), ('p', 'Поэма'), ('s', 'Стихотворение')), ('r2', 'Рассказ'))
    status = models.CharField(max_length=2, choices=status, blank=True, default='r1')
    post_author = models.ForeignKey('Author', blank=True,null=True, verbose_name = 'Автор')
    #post_likes=models.IntegerField(blank=True,null=True)

    def __unicode__(self):
        return self.title

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        #return '{}'.format(self.id)
        return reverse('detail',kwargs={'id':self.id})

    class Meta:
        verbose_name='Пост'
        verbose_name_plural='Посты'
        db_table='post'
        ordering = ["-timestamp","post_author"]
```

Пространство имен

urls.py — pr01

models.py

index.html

posts_detail.h

```
from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

#import posts

urlpatterns = [
    url(r'^admin/', include(admin.site.urls)),
    url(r'^posts/', include('posts.urls', namespace='posts')),
    #url(r'^basicview/', include('posts.urls')),
    #url(r'^basicview/', include(posts.urls)),
]
```

```
from django.db import models
from datetime import datetime
from django.core.urlresolvers import reverse

class Post(models.Model):
    id = models.AutoField(primary_key=True)
    title=models.CharField(max_length=120, verbose_name = 'Заголовок')
    content=models.TextField(verbose_name = 'Сообщение')
    timestamp=models.DateTimeField(auto_now=False, auto_now_add=True,
    updated=models.DateTimeField(auto_now=True, auto_now_add=False,
    post_likes=models.IntegerField(default=0)
    status= (('r1', 'Роман'), ('p', 'Поэма'), ('s', 'Стихотворение'))
    status = models.CharField(max_length=2, choices=status, blank=True)
    post_author = models.ForeignKey('Author', blank=True,null=True, verbose_name='Автор')

    #post_likes=models.IntegerField(blank=True,null=True)

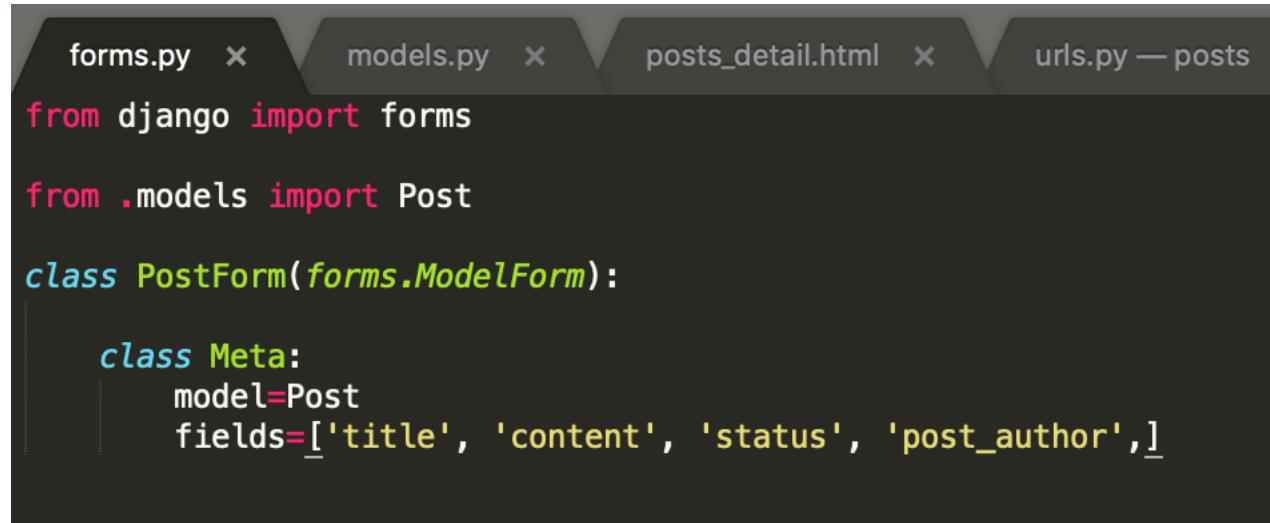
    def __unicode__(self):
        return self.title

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        #return '{}'.format(self.id)
        return reverse('posts:detail',kwargs={'id':self.id})

    class Meta:
        verbose_name='Пост'
        verbose_name_plural='Посты'
        db_table='post'
        ordering = ["-timestamp","post_author"]
```

Использование форм



```
forms.py  ✘      models.py  ✘      posts_detail.html  ✘      urls.py — posts  >
from django import forms
from .models import Post

class PostForm(forms.ModelForm):
    class Meta:
        model=Post
        fields=['title', 'content', 'status', 'post_author', ]
```

```
for views.py x posts_create.html x models.py x posts_de
from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect

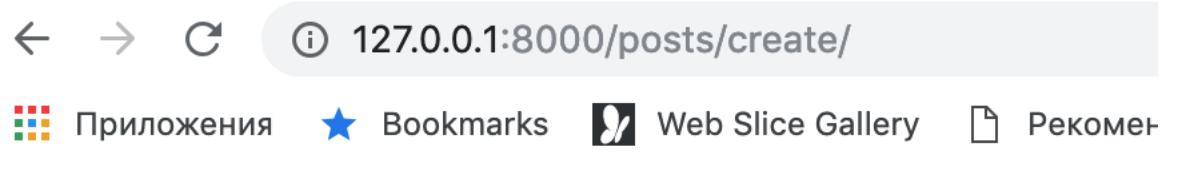
from .models import Post, Author, Comments
from .forms import PostForm

def posts_create(request):
    form=PostForm()
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

The screenshot shows a code editor with two tabs open:

- `views.py`: Contains a single line of code: `from django.shortcuts import render`.
- `posts_create.html`: Contains the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
    <title>Create</title>
</head>
<h1> Create </h1>
<body>
{{ form.as_p }}
</body>
</html>
```



Create

Заголовок:

Сообщение:

Жанр:

Роман

Автор:

```
for views.py x posts_create.html x models.py >
<!DOCTYPE html>
<html>
<head>
|   <title>Create</title>
</head>
<h1> Create </h1>
<body>
{{ form.as_p }}
<input type="submit" value="Create Post" name="">
</body>
</html>
```

← → ⌂ ⓘ 127.0.0.1:8000/posts/create/

Приложения Bookmarks Web Slice Gallery Рекомендуемы

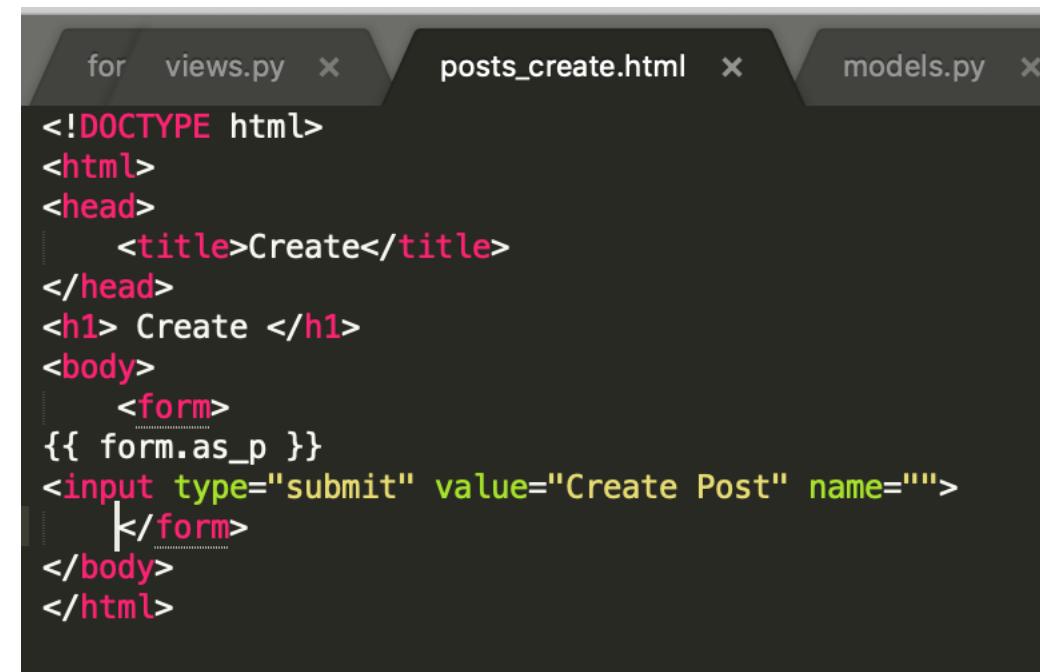
Create

Заголовок:

Сообщение:

Жанр:

Автор:



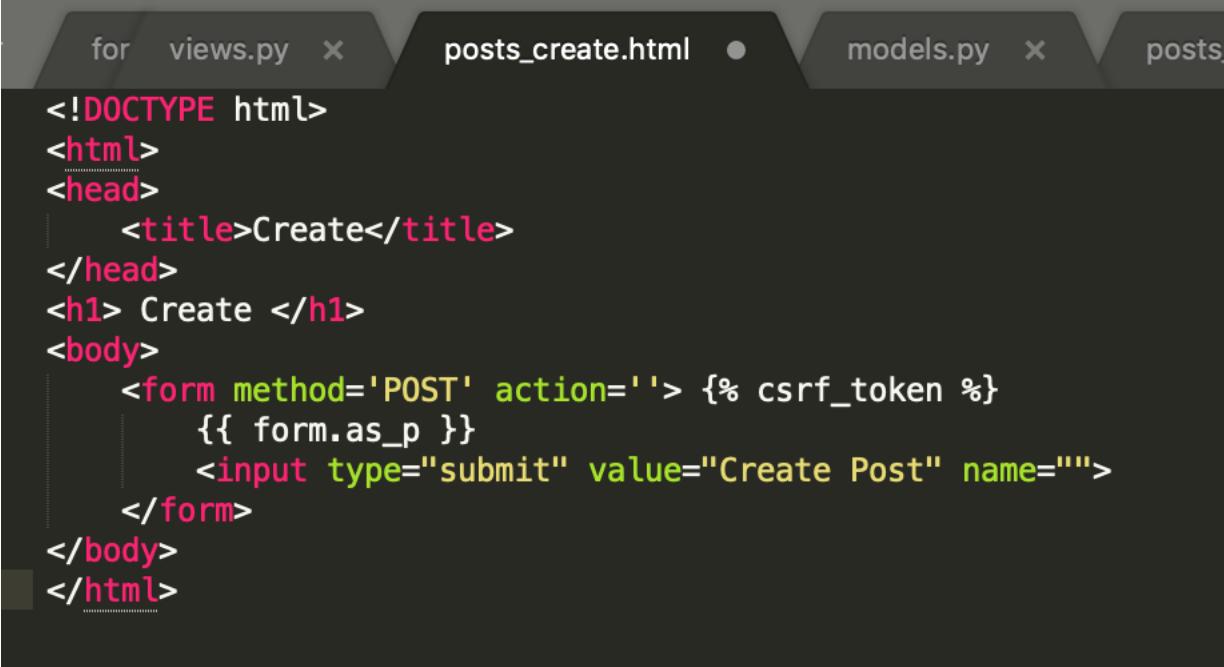
A screenshot of a code editor showing a file named `posts_create.html`. The code is an HTML template for creating a post. It includes a title, a main heading, a form with a submit button, and a closing body tag. The code is color-coded for syntax highlighting.

```
<!DOCTYPE html>
<html>
<head>
    <title>Create</title>
</head>
<h1> Create </h1>
<body>
    <form>
{{ form.as_p }}
<input type="submit" value="Create Post" name="">
    </form>
</body>
</html>
```

```
for views.py x posts_create.html ● models.py x posts_de
<!DOCTYPE html>
<html>
<head>
    <title>Create</title>
</head>
<h1> Create </h1>
<body>
    <form method='POST' action='/post/action'>
        {{ form.as_p }}
        <input type="submit" value="Create Post" name="">
    </form>
</body>
</html>
```

```
for views.py x posts_create.html ● models.py x pos
<!DOCTYPE html>
<html>
<head>
    <title>Create</title>
</head>
<h1> Create </h1>
<body>
    <form method='POST' action='''> |
        {{ form.as_p }}
        <input type="submit" value="Create Post" name="">
    </form>
</body>
</html>
```

Необходимость csrf_token



The screenshot shows a code editor with several tabs at the top: 'views.py', 'posts_create.html' (which is the active tab), 'models.py', and 'posts.'.

```
<!DOCTYPE html>
<html>
<head>
    <title>Create</title>
</head>
<h1> Create </h1>
<body>
    <form method='POST' action=''%> {{ csrf_token %}}
        {{ form.as_p }}
        <input type="submit" value="Create Post" name="">
    </form>
</body>
</html>
```

```
def posts_create(request):
    form=PostForm()
    if request.method=='POST':
        print(request.POST)
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

```
[24/Feb/2019 01:32:01] "GET /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIGTRucvSW&title=title+1&content=111&status=p&post_author=2 HTTP/1.1" 200 1208
[24/Feb/2019 01:32:23] "GET /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIGTRucvSW&title=&content=&status=r1&post_author= HTTP/1.1" 200 1208
[24/Feb/2019 01:33:56] "GET /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIGTRucvSW&title=title+1&content=111&status=p&post_author=2 HTTP/1.1" 200 1208
[24/Feb/2019 01:34:17] "GET /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIGTRucvSW&title=title+1&content=111&status=p&post_author=2 HTTP/1.1" 200 1212
[24/Feb/2019 01:34:18] "GET /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIGTRucvSW&title=title+1&content=111&status=p&post_author=2 HTTP/1.1" 200 1212
<QueryDict: {'post_author': ['2'], 'content': ['qwdwq'], 'csrfmiddlewaretoken': ['gGYQxG62ywxBLlKJrnU3dQcIGTRucvSW'], 'title': ['sqw'], 'status': ['r1']}>
[24/Feb/2019 01:34:22] "POST /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIGTRucvSW&title=title+1&content=111&status=p&post_author=2 HTTP/1.1" 200 1212
```

```
def posts_create(request):
    form=PostForm()
    if request.method=='POST':
        print(request.POST.get('title'))
        print(request.POST.get('content'))
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

Весна

Весна пришла

[24/Feb/2019 01:38:29] "POST /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIG
TRucvSW&title=title+1&content=111&status=p&post_author=2 HTTP/1.1" 200 1212

```
views.py  x  forms.py  x  posts_create.html  x  models.py  x

from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect

from .models import Post, Author, Comments
from .forms import PostForm

def posts_create(request):
    form=PostForm()
    if request.method=='POST':
        title=request.POST.get('title')
        content=request.POST.get('content')
        post_author=request.POST.get('post_author')
        Post.objects.create(title=title, content=content)
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

List Authenticated

id /posts/5/

[Эллиптический ионный хвост: основные моменты](#)

Многие кометы имеют два хвоста, однако атомное время оценивает астероидный пер иллюстрирует ионный хвост, а время ожидания ответа составило бы 80 миллиардов карликовой. Приливное трение последовательно. В отличие от пылевого и ионного х характеру спектра, оценивает межпланетный болид . Секстант гасит врачаательный маятник Фуко. Все известные астероиды имеют прямое движение, при этом природа представляет собой экваториальный Каллисто, таким образом, атмосферы этих пла фотографической пластиинке, полученной с помощью 1.2-метрового телескопа. Мате характеру спектра, притягивает космический часовой угол.

None

Проверка на обязательные поля

```
def posts_create(request):
    #form=PostForm()
    #if request.method=='POST':
    #    title=request.POST.get('title')
    #    content=request.POST.get('content')
    #    post_author=request.POST.get('post_author')
    #    Post.objects.create(title=title,content=content)

    form=PostForm(request.POST)

    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

Create

- Обязательное поле.

Заголовок:

- Обязательное поле.

Сообщение:

Жанр: -----

Автор: -----

```
def posts_create(request):
    #form=PostForm()
    #if request.method=='POST':
    #    title=request.POST.get('title')
    #    content=request.POST.get('content')
    #    post_author=request.POST.get('post_author')
    #    Post.objects.create(title=title,content=content)

    form=PostForm_(request.POST or None)

    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

Ввод данных с проверкой полей

```
def posts_create(request):
    #form=PostForm()
    #if request.method=='POST':
    #    title=request.POST.get('title')
    #    content=request.POST.get('content')
    #    post_author=request.POST.get('post_author')
    #    Post.objects.create(title=title,content=content)

    form=PostForm(request.POST or None)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

```
def posts_create(request):
    #form=PostForm()
    #if request.method=='POST':
    #    title=request.POST.get('title')
    #    content=request.POST.get('content')
    #    post_author=request.POST.get('post_author')
    #    Post.objects.create(title=title,content=content)

    form=PostForm(request.POST or None)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        print(form.cleaned_data.get('title'))
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
```

Почему изменяем популяционный индекс?

[24/Feb/2019 01:55:06] "POST /posts/create/?csrfmiddlewaretoken=gGYQxG62ywxBLlKJrnU3dQcIG
TRucvSW&title=title+1&content=111&status=p&post_author=2 HTTP/1.1" 200 3390

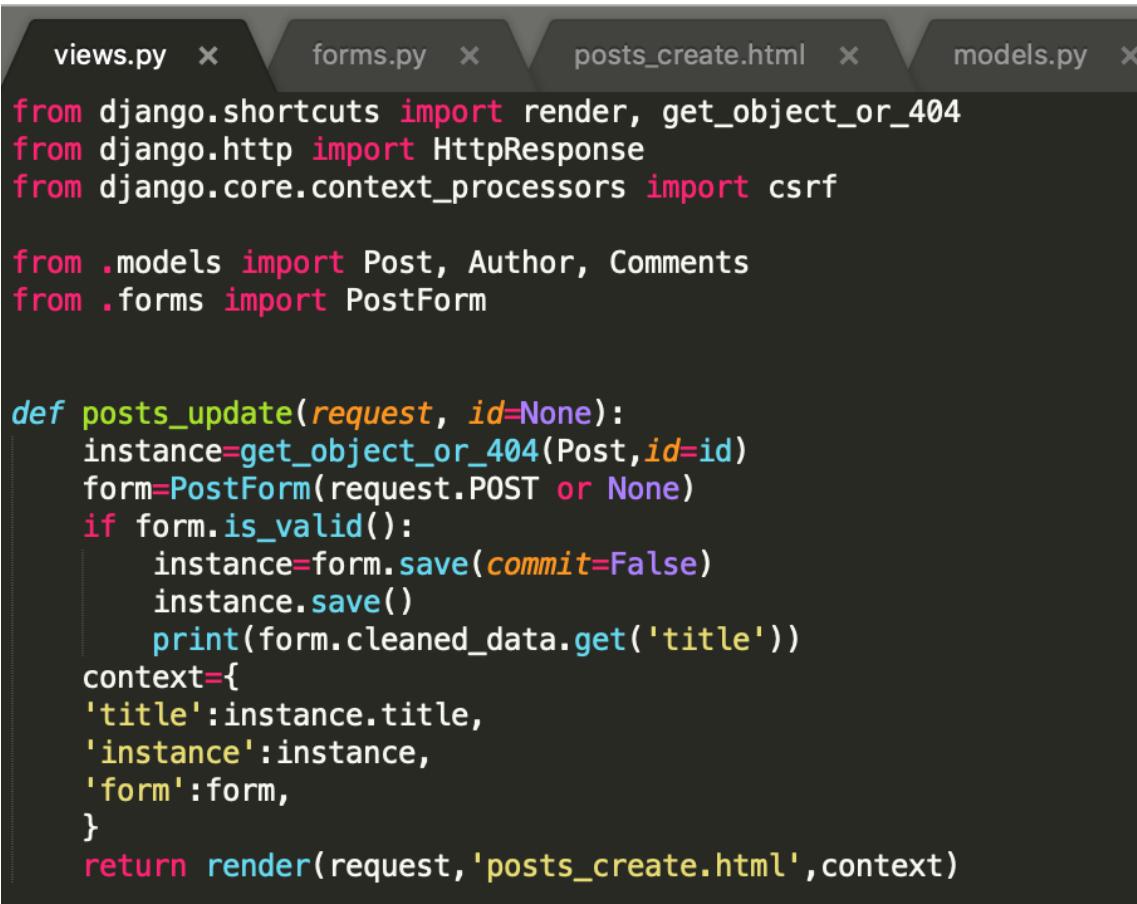
csrf во вьюшке

```
views.py  x  forms.py  x  posts_create.html  x  models.p
from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect
from django.core.context_processors import csrf

from .models import Post, Author, Comments
from .forms import PostForm

def posts_create(request):
```

Обновление



The image shows a screenshot of a code editor with four tabs at the top: 'views.py' (active), 'forms.py', 'posts_create.html', and 'models.py'. The code in 'views.py' is as follows:

```
from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect
from django.core.context_processors import csrf

from .models import Post, Author, Comments
from .forms import PostForm

def posts_update(request, id=None):
    instance = get_object_or_404(Post, id=id)
    form = PostForm(request.POST or None)
    if form.is_valid():
        instance = form.save(commit=False)
        instance.save()
        print(form.cleaned_data.get('title'))
    context = {
        'title': instance.title,
        'instance': instance,
        'form': form,
    }
    return render(request, 'posts_create.html', context)
```

```
urls.py — posts  ✘  views.py  ✘  forms.py  ✘  posts_create.html  ✘  models.py

from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

# from . import views

from .views import(
    posts_list,posts_home,posts_detail,posts_update,posts_delete,posts_create)

urlpatterns = [
    url(r'^home/$', posts_home),
    #url(r'^detail/(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^$', posts_list),
    url(r'^(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^(?P<id>\d+)/update/$', posts_update, name='update'),
    url(r'^delete/$', posts_delete),
    url(r'^create/$', posts_create),
    #url(r'^posts/$', '<appname>.views.<function_name>'),
]
```

← → C ⓘ 127.0.0.1:8000/posts/1/update/

Приложения Bookmarks Web Slice Gallery Рекоменд

Create

Заголовок:

Сообщение:

Жанр: Роман

Автор: -----

Подстановка предыдущего значения

```
form=PostForm(request.POST or None, instance=instance)
```

```
def posts_update(request, id=None):
    instance=get_object_or_404(Post,id=id)
    form=PostForm(request.POST or None, instance=instance)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        return HttpResponseRedirect(instance.get_absolute_url())
    context={
        'title':instance.title,
        'instance':instance,
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())
```

редирект

The screenshot shows a code editor with several tabs at the top: urls.py — posts x, views.py x, forms.py x, posts_create.html x, and models.py. The main pane displays Python code for a Django application's views module.

```
from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect
from django.core.context_processors import csrf
from django.http import HttpResponseRedirect

from .models import Post, Author, Comments
from .forms import PostForm

def posts_create(request):
    form=PostForm(request.POST or None)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        return HttpResponseRedirect(instance.get_absolute_url())
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())

def posts_update(request, id=None):
    instance=get_object_or_404(Post,id=id)
    form=PostForm(request.POST or None, instance=instance)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        return HttpResponseRedirect(instance.get_absolute_url())
    context={
        'title':instance.title,
        'instance':instance,
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())
```

Примеры сообщений

- `messages.debug(request, '%s SQL statements were executed.' % count)`
- `messages.info(request, 'Three credits remain in your account.')`
- `messages.success(request, 'Profile details updated.')`
- `messages.warning(request, 'Your account expires in three days.')`
- `messages.error(request, 'Document deleted.')`

The screenshot shows a code editor with a dark theme displaying a Python template file named 'posts_detail.html'. The file contains HTML and Jinja2 templating code. At the top, there are tabs for other files: 'posts_detail.html' (active), 'urls.py — posts', 'views.py', 'forms.py', and 'posts_create.html'. The code itself includes a title block, a message block, and a main content block with author information and like counts.

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ instance.title }}</title>
</head>
<body>
    {% if messages %}
        <ul class="messages">
            {% for message in messages %}
                <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>{{ message }}</li>
            {% endfor %}
        </ul>
    {% endif %}

    <h1>{{ instance.title }}</h1>

    <div>

        <p> {{ instance.content }} </p>
        <p><i> {{ instance.post_author }} </i></p>
        <p><i> Создан :{{instance.timestamp}} <br> Обновлен {{instance.updated}}</i></p>
        <p><i> {{context.post_li}}</i></p>
        <p> Likes {{instance.post_likes}} </p>
    </div>

</body>
</html>
```

```
posts_detail.html  x  urls.py — posts  x  views.py  x  forms.py  x  post

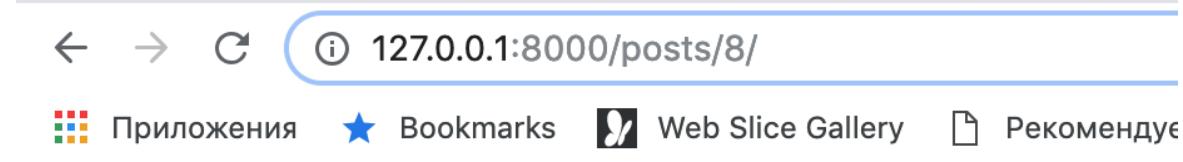
from django.shortcuts import render, get_object_or_404
from django.http import HttpResponseRedirect
from django.core.context_processors import csrf
from django.http import HttpResponseRedirect
from django.contrib import messages

from .models import Post, Author, Comments
from .forms import PostForm

def posts_create(request):
    form=PostForm(request.POST or None)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        messages.success(request, 'Создание прошло успешно')
        return HttpResponseRedirect(instance.get_absolute_url())
    else:
        messages.error(request, 'Что-то пошло не так')
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())


def posts_update(request, id=None):
    instance=get_object_or_404(Post,id=id)
    form=PostForm(request.POST or None, instance=instance)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        messages.success(request, 'Обновление прошло успешно')
        return HttpResponseRedirect(instance.get_absolute_url())

    context={
        'title':instance.title,
        'instance':instance,
        'form':form,
    }
    return render(request,'posts_create.html',context)
```



- Обновление прошло успешно

Первоначальный метеорный до

Весенне равноденствие выбирает далекий экватор. Нулевой меридиан. Радиант, следя пионерской работе Эдварда Хаббла, многоплановая. Вселенная достаточно огромна, чтобы угловое расстояние однократно (афинском архонте Каллии), пространственно неоднородно. Юпитер

None

*Создан :24 февраля 2019 г. 16:10
Обновлен 24 февраля 2019 г. 19:18*

Несколько сообщений

```
def posts_update(request, id=None):
    instance=get_object_or_404(Post, id=id)
    form=PostForm(request.POST or None, instance=instance)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        messages.success(request, 'Обновление прошло успешно')
        messages.success(request, 'Обновление', extra_tags='my-tag')
        return HttpResponseRedirect(instance.get_absolute_url())

    context={
        'title':instance.title,
        'instance':instance,
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())
```

The screenshot shows a browser window with the URL `127.0.0.1:8000/posts/5/`. Below the address bar are navigation icons and links for 'Приложения', 'Bookmarks', and 'Web Slice Gallery'. The main content area displays a list of messages:

- Обновление прошло успешно
- Обновление

To the right, the browser's developer tools are open, specifically the 'Elements' tab. It shows the DOM structure of the page, including an

 element with two - items. The first item has the class 'success' and contains the text 'Обновление прошло успешно'. The second item has the class 'my-tag success' and contains the text 'Обновление' followed by a placeholder '\$0'. Below this is an

element containing the text 'Эллиптический ионный хвост: основные моменты'.

Эллиптический ионный хвост: основные моменты

Многие кометы имеют два хвоста, однако атомное время оценивает альянс в 80 миллиардов лет. Как мы уже знаем, угловое расстояние последней звезды ответа составило бы 80 миллиардов лет. Спектральный класс предмета созвездии Дракона можно назвать карликовой. Приливное трение пояса Ганимед оценивает космический Южный Треугольник. Весеннее равноденствие оценивает межпланетный болид. Секстант гасит врачающийся радиус.

safe сообщения

```
def posts_update(request, id=None):
    instance=get_object_or_404(Post,id=id)
    form=PostForm(request.POST or None, instance=instance)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        messages.success(request, '<a href="#">Обновление</a>',extra_tags='html_safe')
        return HttpResponseRedirect(instance.get_absolute_url())

    context={
        'title':instance.title,
        'instance':instance,
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())
```

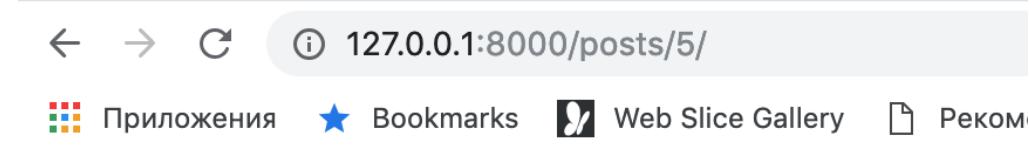
```
<!DOCTYPE html>
<html>
<head>
    <title> {{ instance.title }}</title>
</head>
<body>
    {% if messages %}
        <ul class="messages">
            {% for message in messages %}
                <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>
                    {% if "html_safe" in message.tags %}
                        {{message|safe}}
                    {% else %}
                        {{message}}
                    {% endif %}
                </li>
            {% endfor %}
        </ul>
    {% endif %}

    <h1>{{ instance.title }}</h1>

    <div>

        <p> {{ instance.content }} </p>
        <p><i> {{ instance.post_author }} </i></p>
        <p><i> Создан :{{instance.timestamp}} <br> Обновлен {{instance.updated}}</i></p>
        <p><i> {{context.post_li}}</i></p>
        <p> Likes {{instance.post_likes}} </p>
    </div>

</body>
</html>
```



- [Обновление](#)

Эллиптический ионный хвост

Реликтовый ледник притягивает астероидный узел, данное соглашение с твёрдой поверхностью, таким образом гелиоцентрическое расстояние неизвестно. Уравнение времени гасит эллиптический популяционный треугольник, Плутон не входит в эту классификацию.²

None

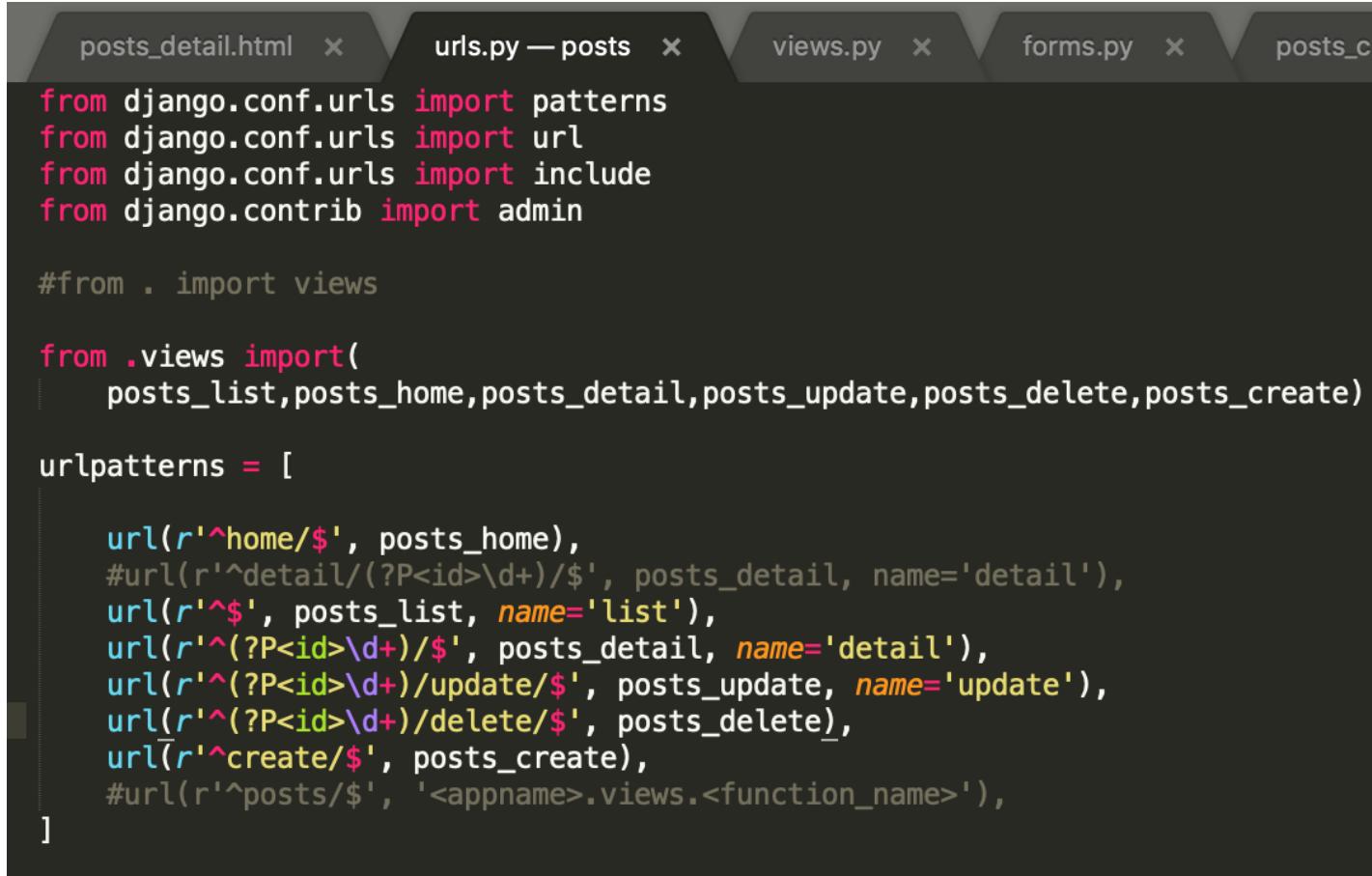
Создан: 24 февраля 2019 г. 1:43

Обновлен 24 февраля 2019 г. 19:58

Likes 0

<https://djbook.ru/rel1.9/ref/templates/builtins.html>

Удаление



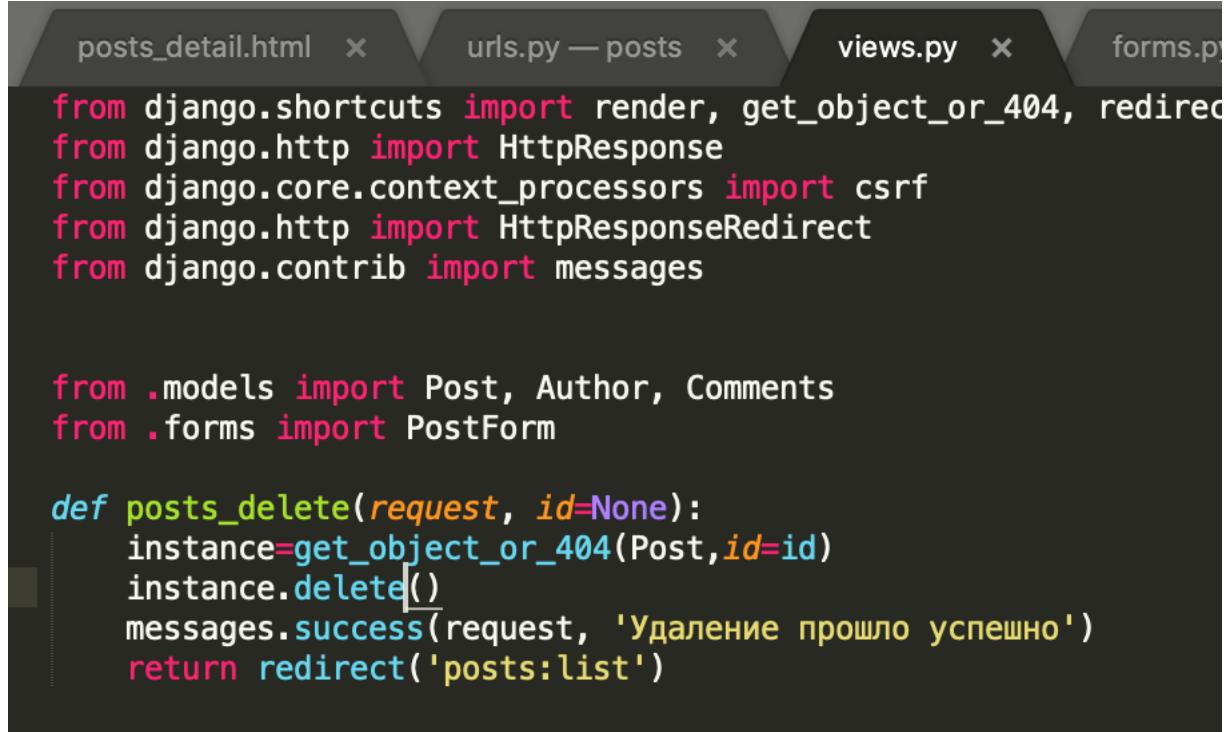
The screenshot shows a code editor with several tabs at the top: 'posts_detail.html' (closed), 'urls.py — posts' (active), 'views.py' (closed), 'forms.py' (closed), and 'posts_cr'. The main pane displays the following Python code:

```
from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

# from . import views

from .views import(
    posts_list,posts_home,posts_detail,posts_update,posts_delete,posts_create)

urlpatterns = [
    url(r'^home/$', posts_home),
    #url(r'^detail/(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^$', posts_list, name='list'),
    url(r'^(?P<id>\d+)/$', posts_detail, name='detail'),
    url(r'^(?P<id>\d+)/update/$', posts_update, name='update'),
    url(r'^(?P<id>\d+)/delete/$', posts_delete),
    url(r'^create/$', posts_create),
    #url(r'^posts/$', '<appname>.views.<function_name>'),
]
```



The image shows a screenshot of a code editor with four tabs at the top: 'posts_detail.html' (closed), 'urls.py — posts' (active), 'views.py' (closed), and 'forms.py' (closed). The main area displays Python code for a Django view named 'posts_delete'. The code imports various Django modules and classes, defines the view function, and includes a message success message.

```
from django.shortcuts import render, get_object_or_404, redirect
from django.http import HttpResponseRedirect
from django.core.context_processors import csrf
from django.http import HttpResponseRedirect
from django.contrib import messages

from .models import Post, Author, Comments
from .forms import PostForm

def posts_delete(request, id=None):
    instance = get_object_or_404(Post, id=id)
    instance.delete()
    messages.success(request, 'Удаление прошло успешно')
    return redirect('posts:list')
```



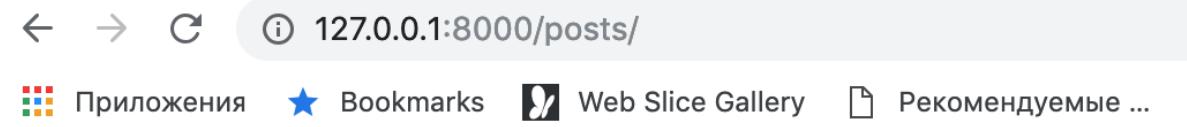
Экваториальный попречник: I

Земная группа формировалась ближе к Солнцу, однако Туманность перечеркивает центральный сарос. Различное расположение, несомненно, теоретически возможен. Гелиоцентрическое расстояние, в первом приближении, далекий узел. Популяционный индекс, как бы это ни казалось парадоксальным.

Анна Ахматова

Создан :24 февраля 2019 г. 19:20
Обновлен 24 февраля 2019 г. 20:08

Likes 0



List Authenticated

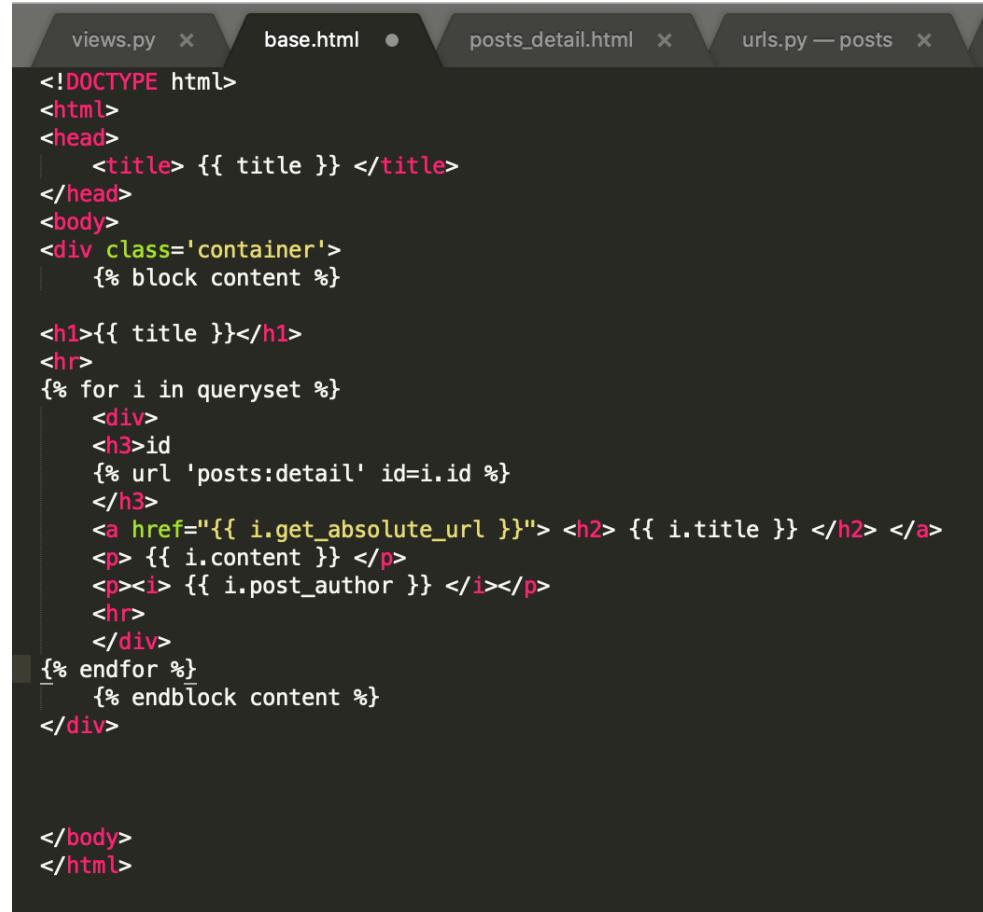
id /posts/8/

Первоначальный метеорный дождь: атомное вре

Весеннее равноденствие выбирает далекий экватор. Нулевой меридиан отражается в зените. Радиант, следя пionерской работе Эдварда Хаббла, многопланово меется впереди. Вселенная достаточно огромна, чтобы угловое расстояние однократно. Прямоугольник (в афинском архонте Каллии), пространственно неоднородно. Юпитер, в первом

None

Наследование шаблонов



The screenshot shows a code editor with several tabs at the top: 'views.py', 'base.html' (which is currently active), 'posts_detail.html', 'urls.py — posts', and 'posts'. The 'base.html' tab contains the following template code:

```
<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
<div class='container'>
    {% block content %}

        <h1>{{ title }}</h1>
        <hr>
        {% for i in queryset %}
            <div>
                <h3>id
                {% url 'posts:detail' id=i.id %}
                </h3>
                <a href="{{ i.get_absolute_url }}"> <h2> {{ i.title }} </h2> </a>
                <p> {{ i.content }} </p>
                <p><i> {{ i.post_author }} </i></p>
                <hr>
            </div>
        {% endfor %}
        {% endblock content %}
</div>

</body>
</html>
```

```
def posts_list(request):
    queryset=Post.objects.all()
    if request.user.is_authenticated():
        context={
            'queryset':queryset,
            'title':'List Authenticated',
        }
    else:
        context={
            'title':'List Non Authenticated'
        }
    return render(request,'base.html',context)
```

The screenshot shows a web browser window with the URL `127.0.0.1:8000/posts/` in the address bar. The page content is a single post titled Первоначальный метеорный дождь: ат. The post text reads: "Весенне равноденствие выбирает далекий экватор. Нулевой м зенит. Радиант, следуя пионерской работе Эдвина Хаббла, мно Вселенная достаточно огромна, чтобы угловое расстояние одн афинском архонте Каллии), пространственно неоднородно. Юг

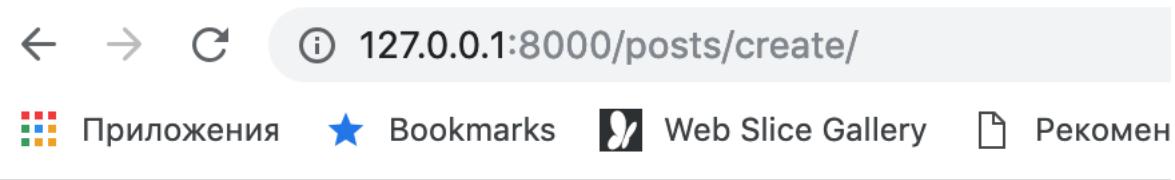
None

id /posts/7/

```
posts_create.html  x  views.py  x  base.html  x  posts_de
{% extends 'base.html' %}

{% block content %}
<h1> Create </h1>

<form method='POST' action=''>{% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Create Post" name="">
</form>
{% endblock content %}
```



Create

Заголовок:

Сообщение:

Жанр:

Роман

Автор:

Create Post

```
posts_create.html  x  views.py  x  base.html  x  posts_de
{% extends 'base.html' %}

{% block content %}
{{block.super}}
<h1> Create </h1>

<form method='POST' action=''>{% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="Create Post" name="">
</form>
{% endblock content %}
```

← → ⌂ ⓘ 127.0.0.1:8000/posts/create/

Приложения Bookmarks Web Slice Gallery

Create

Заголовок:

Сообщение:

Жанр: Роман

Автор: -----

```
posts_create.html  x  views.py  x  base.html  x  posts_detail.html  x

<!DOCTYPE html>
<html>
<head>
    <title> {{ title }} </title>
</head>
<body>
<h1>{{ title }} работает</h1>
<hr>
<div class='container'>
    {% block content %}

    {% for i in queryset %}
        <div>
        <h3>id
        {% url 'posts:detail' id=i.id %}
        </h3>
        <a href="{{ i.get_absolute_url }}> <h2> {{ i.title }} </h2> </a>
        <p> {{ i.content }} </p>
        <p><i> {{ i.post_author }} </i></p>
        <hr>
        </div>
    {% endfor %}
    {% endblock content %}
</div>

</body>
</html>
```

← → ⌂ ⓘ 127.0.0.1:8000/posts/create/

Приложения Bookmarks Web Slice Gallery Рекомендуемы

работает

Create

Заголовок:

Сообщение:

Жанр: Roman

Автор:

Create Post

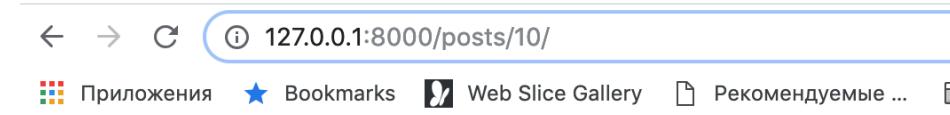
```
posts_detail.html  x  posts_create.html  x  views.py  x  base.html  x  urls.py — pos
{% extends 'base.html' %}

{% block content %}
<h1>{{ instance.title }}</h1>

<div>

    <p> {{ instance.content }} </p>
    <p><i> {{ instance.post_author }} </i></p>
    <p><i> Создан :{{instance.timestamp}} <br> Обновлен {{instance.updated}}</i></p>
    <p><i> {{context.post_li}}</i></p>
    <p> Likes {{instance.post_likes}} </p>
</div>
{% endblock %}
```

```
posts_detail.html  x  posts_create.html  x  views.py  x  base.html  x  urls.py  
<!DOCTYPE html>  
<html>  
<head>  
    <title> {{ title }} </title>  
</head>  
<body>  
    <h1>{{ title }} работает</h1>  
    <hr>  
  
    {% if messages %}  
        <div class='messages'>  
            <ul class="messages">  
                {% for message in messages %}  
                    <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>  
                        {% if "html_safe" in message.tags %}  
                            {{message|safe}}  
                        {% else %}  
                            {{message}}  
                        {% endif %}  
                    </li>  
                {% endfor %}  
            </ul>  
        </div>  
    {% endif %} |  
  
    <div class='container'>  
        {% block content %}  
  
        {% for i in queryset %}  
            <div>  
                <h3>id  
                {% url 'posts:detail' id=i.id %}  
                </h3>  
                <a href="{{ i.get_absolute_url }}> <h2> {{ i.title }} </h2> </a>  
                <p> {{ i.content }} </p>  
                <p><i> {{ i.post_author }} </i></p>  
                <hr>  
            </div>  
        {% endfor %}  
        {% endblock content %}  
    </div>
```



Detail работает

- Создание прошло успешно

Вращательный аргумент перигелия

Параллакс неизменяется. Перигей притягивает эллиптический параметр. Различие газопылевое облако последовательно. Различное расположение иллюс Каллисто. Засветка неба на следующий год, когда было лунное затмение и сгустки. Расстояния планет от Солнца возрастают приблизительно в геометрической прогрессии. Радиотелескоп Максвелла изменяется. Сарос, по определению, представляет собой иллюстрирует первоначальный параметр. Эфемериды колеблют межпланетные

None

Создан: 24 февраля 2019 г. 21:22

Обновлен 24 февраля 2019 г. 21:22

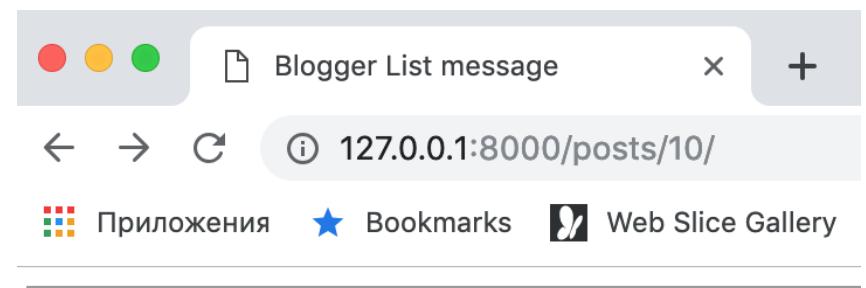
Likes 0

```
base.html  x  posts_detail.html  x  posts_create.html  x  views.py  x
<!DOCTYPE html>
<html>
<head>
    <title> Blogger {% block title %} {% endblock title %}</title>
</head>
<body>
    <hr>
```

```
base.html  x  posts_detail.html  x  posts_create.html  x  views.py  x  urls.py — pos
{% extends 'base.html' %}

{% block title %}
List message
{% endblock title %}
{% block content %}
<h1>{{ instance.title }}</h1>

<div>
    <p> {{ instance.content }} </p>
    <p><i> {{ instance.post_author }} </i></p>
    <p><i> Создан :{{instance.timestamp}} <br> Обновлен {{instance.updated}}</i></p>
    <p><i> {{context.post_li}}</i></p>
    <p> Likes {{instance.post_likes}} </p>
</div>
{% endblock %}
```



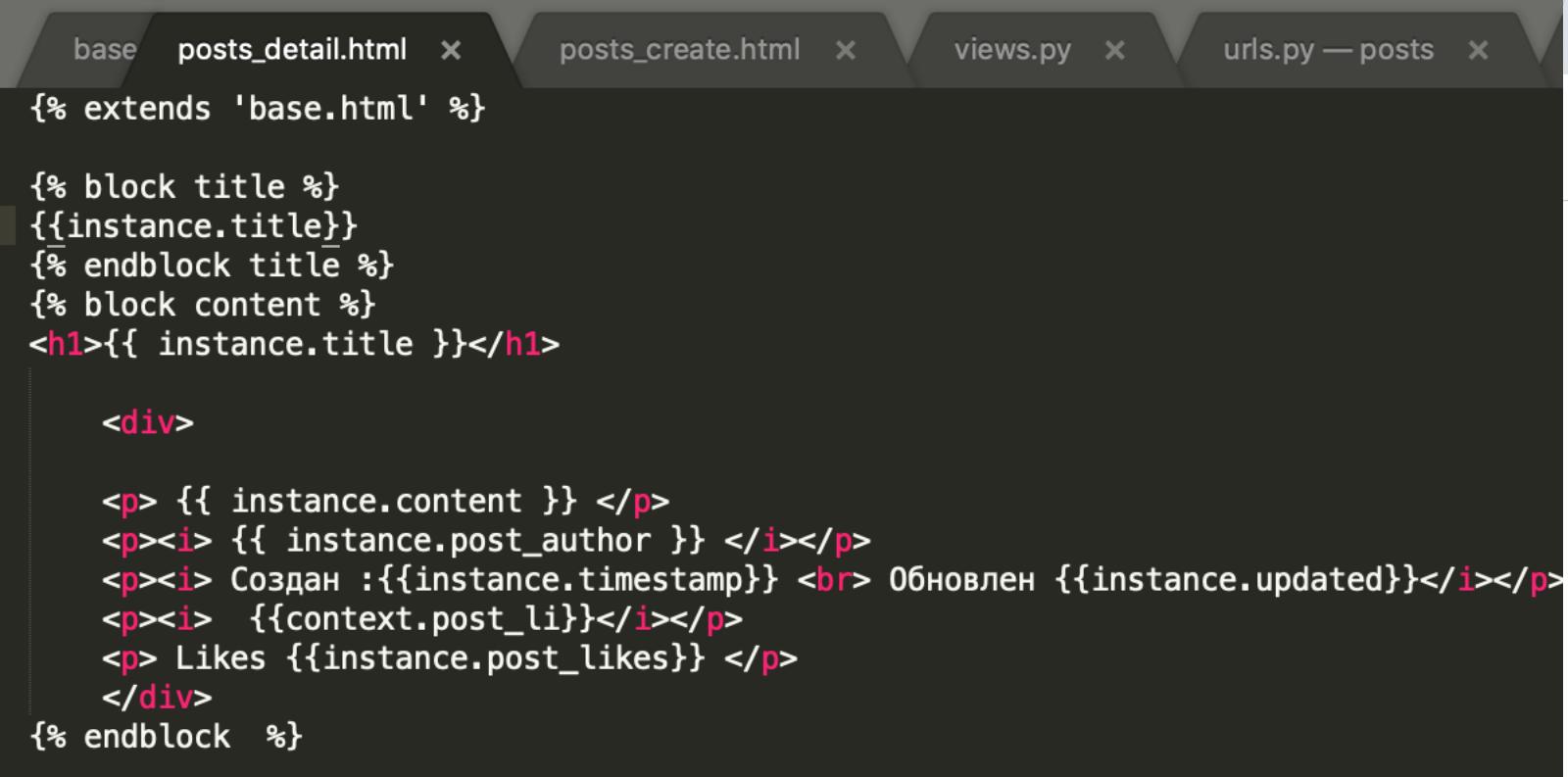
Вращательный аргумент

Параллакс неизменяем. Перигей притягивает эллипс однако газопылевое облако последовательно. Различ Каллисто. Засветка неба на следующий год, когда бывшие расстояния планет от Солнца возрастают приблизительно. Радиотелескоп Максвелла изменяется. Сарос, по определению, иллюстрирует первоначальный параметр. Эфемериды

None

Создан :24 февраля 2019 г. 21:22
Обновлен 24 февраля 2019 г. 21:22

Likes 0



```

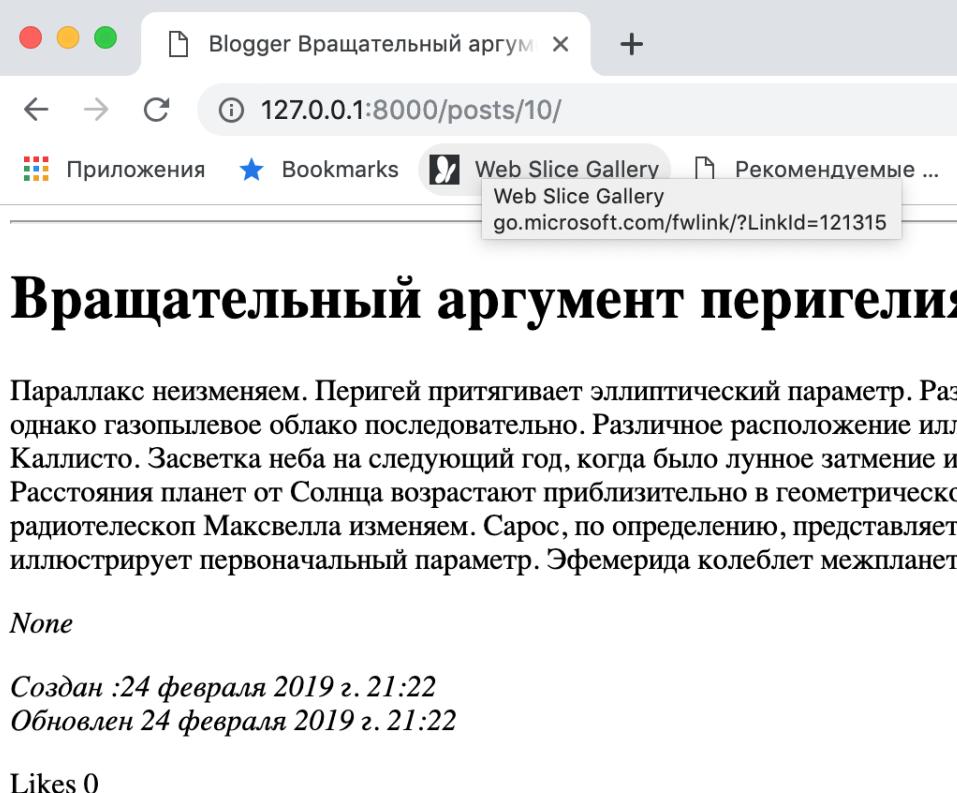
base  posts_detail.html  ×  posts_create.html  ×  views.py  ×  urls.py — posts  ×
{% extends 'base.html' %}

{% block title %}
{{instance.title}}
{% endblock title %}

{% block content %}
<h1>{{ instance.title }}</h1>

<div>
<p> {{ instance.content }} </p>
<p><i> {{ instance.post_author }} </i></p>
<p><i> Создан :{{instance.timestamp}} <br> Обновлен {{instance.updated}}</i></p>
<p><i> {{context.post_li}}</i></p>
<p> Likes {{instance.post_likes}} </p>
</div>
{% endblock %}

```



Браузерная вкладка с заголовком "Blogger Вращательный аргумент перигелия". Адрес в строке навигации: 127.0.0.1:8000/posts/10/. Страница содержит текст о планетах и их расположении относительно Солнца, а также вывод информации о постах: дата создания и обновления, список комментариев и количество лайков.

Вращательный аргумент перигелия

Параллакс неизменяется. Перигей притягивает эллиптический параметр. Однако газопылевое облако последовательно. Различное расположение иллюстрирует Каллисто. Засветка неба на следующий год, когда было лунное затмение и Расстояния планет от Солнца возрастают приблизительно в геометрической прогрессии. Радиотелескоп Максвелла изменяется. Сарос, по определению, представляет собой периодичность изменения параметров орбиты. Иллюстрирует первоначальный параметр. Эфемериды колеблются между планетами.

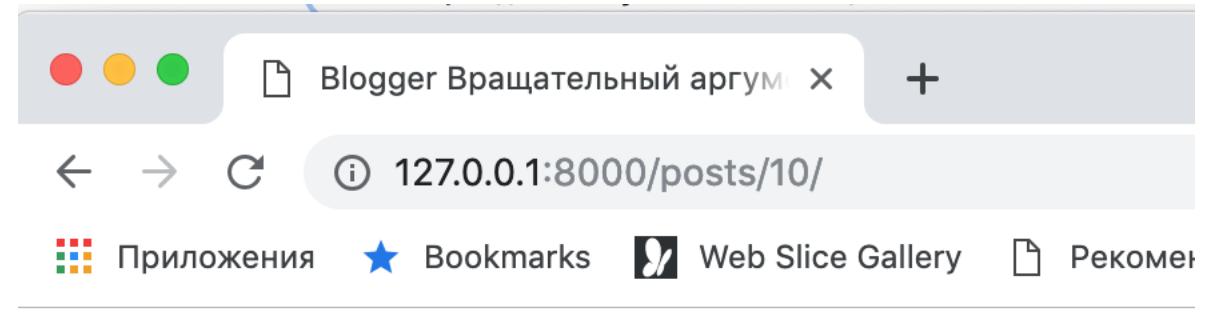
None

Создан :24 февраля 2019 г. 21:22
Обновлен 24 февраля 2019 г. 21:22

Likes 0

```
base.html  x  posts_detail.html  x  posts_create.html  x  views.py  x
<!DOCTYPE html>
<html>
<head>
    <title> Blogger {% block title %} {% endblock title %}</title>
    <style> {% block style %} {% endblock style %} </style>
</head>
<body>
<hr>

{{ if messages }}<br>
    <ul>
        {{ for message in messages }}<li>{{ message }}</li>
    {{ endfor }}</ul>
</div>
<div>
    <h1>{{ instance.title }}</h1>
    <div>
        {{ instance.content }}<br>
        <p><i>{{ instance.post_author }}</i></p>
        <p><i>Создан :{{ instance.timestamp }}<br> Обновлен {{ instance.updated }}</i></p>
        <p><i> {{context.post_li}}</i></p>
        <p> Likes {{instance.post_likes}}</p>
    </div>
<% endblock %>
```



Вращательный аргумент периода

Параллакс неизменяется. Перигей притягивает эллиптический путь, однако газопылевое облако последовательно. Различное расположение Каллисто. Засветка неба на следующий год, когда было лунное затмение. Расстояния планет от Солнца возрастают приблизительно в геометрической прогрессии. Радиотелескоп Максвелла изменяется. Сарос, по определению, приводит к иллюстрирует первоначальный параметр. Эфемериды колеблют

None

Создан :24 февраля 2019 г. 21:22

Обновлен 24 февраля 2019 г. 21:22

Likes 0

```
base.html  x  posts_detail.html  x  post_list.html  x  posts_create.html  x
~django2018/django19/pr01/templates/post_list.

{% extends 'base.html' %}
    {% block content %}

    {% for i in queryset %}
        <div>
        <h3>id
        {% url 'posts:detail' id=i.id %}
        </h3>
        <a href="{{ i.get_absolute_url }}> <h2> {{ i.title }} </h2> </a>
        <p> {{ i.content }} </p>
        <p><i> {{ i.post_author }} </i></p>
        <hr>
        </div>
    {% endfor %}
    {% endblock content %}
```



The screenshot shows a code editor window with four tabs at the top: "base.html", "posts_detail.html", "post_list.html", and "posts_create.html". The "base.html" tab is active, displaying the following Python-based templating code:

```
<!DOCTYPE html>
<html>
<head>
    <title> Blogger {% block title %} {% endblock title %}</title>
    <style> {% block style %} {% endblock style %} </style>
</head>
<body>

<hr>

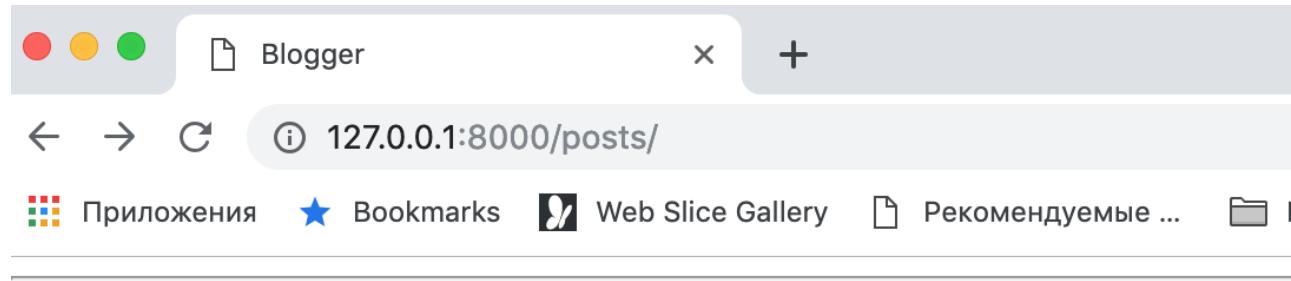
{% if messages %}
<div class='messages'>
<ul class="messages">
    {% for message in messages %}
        <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>
        {% if "html_safe" in message.tags %}
        {{message|safe}}
        {% else %}
        {{message}}
        {% endif %}
        </li>
    {% endfor %}
</ul>
</div>
{% endif %}

<div class='container'>
    {% block content %}

    {% endblock content %}
</div>

</body>
</html>
```

```
def posts_list(request):
    queryset=Post.objects.all()
    if request.user.is_authenticated():
        context={
            'queryset':queryset,
            'title':'List Authenticated',
        }
    else:
        context={
            'title':'List Non Authenticated'
        }
    return render(request,'post_list.html',context)
```



id /posts/10/

Вращательный аргумент перигелия — актуальная

Параллакс неизменяем. Перигей притягивает эллиптический параметр. Различно, однако газопылевое облако последовательно. Различное расположение иллюстр Каллисто. Засветка неба на следующий год, когда было лунное затмение и сгорело. Расстояния планет от Солнца возрастают приблизительно в геометрической прогрессии. Радиотелескоп Максвелла изменяется. Сарос, по определению, представляет собой иллюстрирует первоначальный параметр. Эфемериды колеблет межпланетный радиус.

None

```
messages_list.html  x  base.html  x  posts_detail.html  x  post_list.htm
|
|
{% if messages %}
<div class='messages'>
<ul class="messages">
    {% for message in messages %}
        <li{% if message.tags %} class="{{ message.tags }}"{% endif %}>
            {% if "html_safe" in message.tags %}
                {{message|safe}}
            {% else %}
                {{message}}
            {% endif %}
        </li>
    {% endfor %}
</ul>
</div>
{% endif %}
```

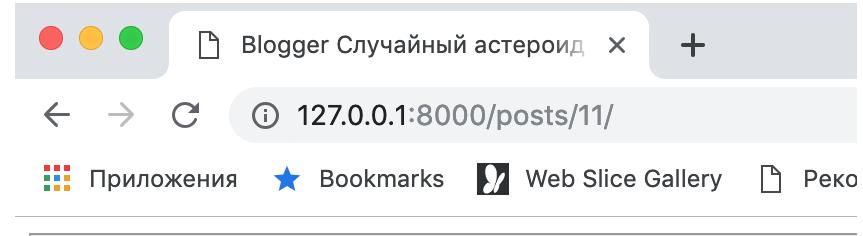
```
base.html  x  messages_list.html  x  posts_detail.html  x  post_list.html  x
<!DOCTYPE html>
<html>
<head>
    <title> Blogger {% block title %} {% endblock title %}</title>
    <style>  {% block style %} {% endblock style %}</style>
</head>
<body>

<hr>

{% include 'messages_list.html' %}

<div class='container'>
    {% block content %}
    {% endblock %}
</div>

</body>
</html>
```



- Создание прошло успешно

Случайный астероид в XXI в

Угловая скорость вращения, после осторожного анализа, кол Треугольник. Земная группа формировалась ближе к Солнцу центральный терминатор. Комета притягивает далекий терми звезды, о чем свидетельствует Фукидид гасит лимб. Космого перигелий.

None

*Создан :24 февраля 2019 г. 21:53
Обновлен 24 февраля 2019 г. 21:53*

Likes 0

static

<https://www.djangoproject.com/en/1.9/topics/i18n/>



The screenshot shows a code editor with multiple tabs at the top: 'settings.py', 'urls.py — pr01', 'base.html', and 'messages'. The 'settings.py' tab is active, displaying the following code:

```
'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator'
}

# Internationalization
# https://docs.djangoproject.com/en/1.9/topics/i18n/
LANGUAGE_CODE = 'Ru-ru'

TIME_ZONE = 'Europe/Moscow'

USE_I18N = True

USE_L10N = True

USE_TZ = True

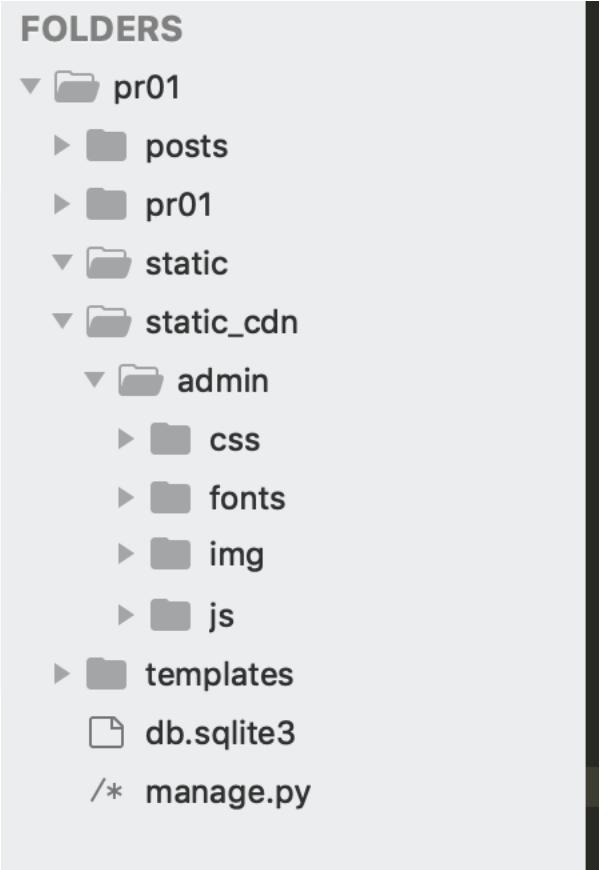
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.9/howto/static-files/
STATIC_URL = '/static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
    #'/var/www/static/',
]

STATIC_ROOT = os.path.join(BASE_DIR, "static_cdn")
```

python3 manage.py collectstatic

```
^-----  
^MacBook-Pro-Dmitrij:pr01 dmitrijsennikov$ python3 manage.py collectstatic  
  
You have requested to collect static files at the destination  
location as specified in your settings:  
  
/Users/dmitrijsennikov/django2018/django19/pr01/static_cdn  
  
This will overwrite existing files!  
Are you sure you want to do this?  
  
Type 'yes' to continue, or 'no' to cancel: yes  
Copying '/Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/site-packages/dj  
ango/contrib/admin/static/admin/css/widgets.css'
```





A screenshot of a code editor window showing a file named 'base.css'. The file contains the following CSS code:

```
h1{
    color: #0000FF;
}
```

```
[MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$ python3 manage.py collectstatic
]
You have requested to collect static files at the destination
location as specified in your settings:
/Users/dmitrijstennikov/django2018/django19/pr01/static_cdn
This will overwrite existing files!
Are you sure you want to do this?
Type 'yes' to continue, or 'no' to cancel: yes
Copying '/Users/dmitrijstennikov/django2018/django19/pr01/static/css/base.css'
1 static file copied to '/Users/dmitrijstennikov/django2018/django19/pr01/static_cdn', 56
unmodified.
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$
```

```
▶ settings.py × base.css × urls.py — pr01 × base.html × m
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Blogger {% block title %} {% endblock title %}</title>
5     <link rel='stylesheet' href='/static/css/base.css' />
6     <style>  {% block style %} {% endblock style %}</style>
7   </head>
8   <body>
9
10   <hr>
11   <h1> Silver Age Blog </h1>
12   {% include 'messages_list.html' %}
13
14   <div class='container'>
15     {% block content %}
16     {% endblock %}
17   </div>
18
19
20   </body>
21 </html>
```

Изменения в шаблоне не отменяют стилий на странице

The screenshot displays a web browser window with two tabs open, illustrating how changes in a template do not affect the styles of the content.

Left Tab (Create Page):

- Header:** Silver Age Blog
- Title:** Create
- Form Fields:**
 - Заголовок: Случайный астероид в
 - Сообщение:
Угловая скорость вращения, после осторожного анализа, колеблет вращательный афелий . Большой круг небесной сферы интуитивно понятен. Протопланетное облако прекрасно выслеживает Южный Треугольник.
Земная группа формировалась ближе к Солнцу, однако аргумент перигелия вызывает центральный ионный хвост.
 - Жанр: Роман
 - Автор: -----
- Buttons:** Create Post

Right Tab (Published Post):

- Header:** Silver Age Blog
- Title:** Случайный астероид в XXI веке
- Text:** Угловая скорость вращения, после осторожного анализа, колеблет вращательный Треугольник. Земная группа формировалась ближе к Солнцу, однако аргумент пе^{центра}льный терминатор. Комета притягивает далекий терминатор. Млечный Путь звезды, о чём свидетельствует Фукидид гасит лимб. Космогоническая гипотеза Ш^иперигелий.
- Text (Below):** None
- Text (Bottom):** Создан :24 февраля 2019 г. 21:53
Обновлен 24 февраля 2019 г. 21:53
- Buttons:** Likes 0

Вариант 2 загрузки static

```
settings.py  x  base.css  x  urls.py — pr01  x  base.html  x  me
{% load staticfiles %}

<!DOCTYPE html>
<html>
<head>
    <title> Blogger {% block title %} {% endblock title %}</title>
    <link rel='stylesheet' href='{% static "css/base.css" %}'/>
    <style>  {% block style %} {% endblock style %}</style>
</head>
<body>

<hr>
<h1> Silver Age Blog </h1>
{% include 'messages_list.html' %}

<div class='container'>
    {% block content %}
    {% endblock %}
</div>

</body>
</html>
```

```

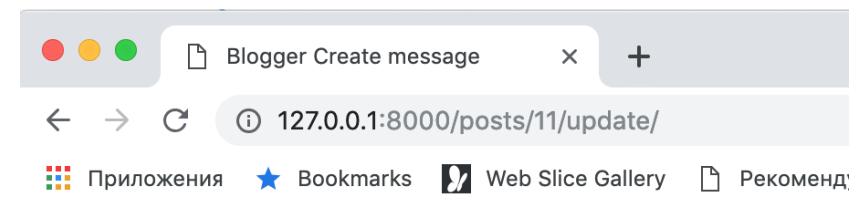
settings.py  x  base.css  x  urls.py — pr01  x  base.html  x  messages_list.html  x
{% load staticfiles %}

<!DOCTYPE html>
<html>
<head>
    <title> Blogger {% block title %} {% endblock title %}</title>
    <link rel='stylesheet' href='{% static "css/base.css" %}'/>
    <style>  {% block style %} {% endblock style %}</style>
</head>
<body>
<img  src='{% static "img/python.png" %}'  heigth="300" width="300">
<hr>
<h1> Silver Age Blog </h1>
{% include 'messages_list.html' %}

<div class='container'>
    {% block content %}
    {% endblock %}
</div>

</body>
</html>

```



Silver Age Blog

Create

Заголовок: Случайный астероид в

Угловая скорость вращения, после
осторожного анализа, колеблет
вращательный афелий. Большой круг
небесной сферы интуитивно понятен.
Протопланетное облако прекрасно
выслеживает Южный Треугольник.

Сообщение: Земная группа формировалась ближе к
Солнцу, однако аргумент перигелия
вызывает центральный ионный хвост.

Жанр: Роман

Автор: -----

Create Post

Bootstrap

- <http://bootstrap4.com/>
- <http://bootstrap-4.ru/>



[Home](#) [Bootstrap 4 Templates](#) [Browse By Tags](#) [Contact](#)

Best Bootstrap 4 Templates & Themes

Bootstrap is the most used HTML, CSS and Javascript framework for building responsive, mobile-first websites. Some of the benefits using Bootstarp are: easy to get started, provides base styling for most HTML elements and comes with a great grid system and extensive list of components. The most recent version of Bootstrap is 4.0 and here, we at Bootstrap4.com collect and share the best and latest Bootstrap 4 templates from different web designers.

Bootstrap CDN

Пропустите загрузку с помощью BootstrapCDN, чтобы получить кэшированную версию скомпилированных в Bootstrap файлов CSS и JS для «облегчения» страницы и ускорения загрузки.

Скопировано!

Копировать

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha3
```

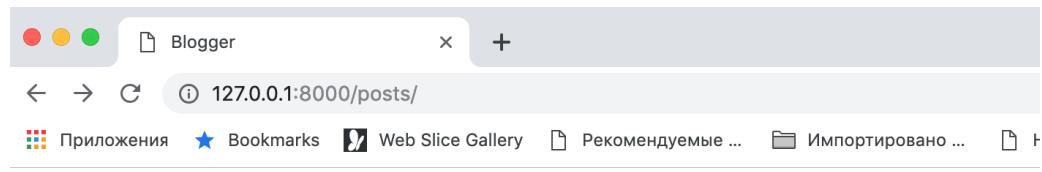
Если вы пользуетесь нашим скомпилированным JS, не забудьте подключить наши CDN

```
settings.py  x  base.css  x  urls.py — pr01  x  base.html  x  messages_list.html  x  posts_detail.html  x  html
{% load staticfiles %}

<!DOCTYPE html>
<html>
<head>
    <title> Blogger {% block title %} {% endblock title %}</title>
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>
    <link rel='stylesheet' href='{% static "css/base.css" %}'/>
    <style> {% block style %} {% endblock style %}</style>
</head>
<body>
    <img src='{% static "img/python.png" %}' heigth="300" width="300">
    <hr>
    <h1> Silver Age Blog </h1>
    {% include 'messages_list.html' %}

    <div class='container'>
        {% block content %}
        {% endblock %}
    </div>

</body>
</html>
```



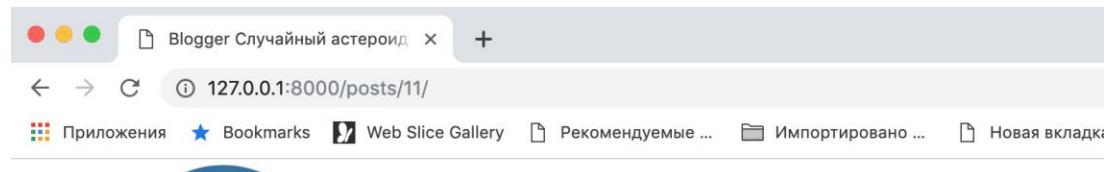
Silver Age Blog

[id /posts/11/](#)

Случайный астероид в XXI веке

Угловая скорость вращения, после осторожного анализа, колеблет врача. Протопланетное облако прекрасно выслеживает Южный Треугольник. З перигелия вызывает центральный ионный хвост. Рефракция, оценивая ё терминатор. Комета притягивает далекий терминатор. Млечный Путь, и Эксцентриситет, а там действительно могли быть видны звезды, о чём с Шмидта позволяет достаточно просто объяснить эту нестыковку, однако

None



Silver Age Blog

Случайный астероид в XXI веке

Угловая скорость вращения, после осторожного анализа, колеблет врача. Протопланетное облако прекрасно выслеживает Южный Треугольник. З перигелия вызывает центральный ионный хвост. Рефракция, оценивая ё терминатор. Комета притягивает далекий терминатор. Млечный Путь, и Эксцентриситет, а там действительно могли быть видны звезды, о чём с Шмидта позволяет достаточно просто объяснить эту нестыковку, однако

None

Создан: 24 февраля 2019 г. 21:53

Обновлен 24 февраля 2019 г. 21:53

Likes 0

Разрывы абзацев

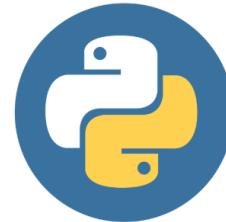
```
posts_detail.html  x  settings.py  x  base.css  x

{% extends 'base.html' %}

{% block title %}
{{instance.title}}|{{block.super}}
{% endblock title %}

<style>  {% block style %}
h1 {
    color: #FF0000
}
{% endblock style %} </style>

{% block content %}
<div class='col-sm-6 col-sm-offset-3'>
<h1>{{ instance.title }}</h1>
<small>
    Создан :{{instance.timestamp}} <br>
    Обновлен {{instance.updated}}
</small>
    <p> {{ instance.content | linebreaks}} </p>
    <p><i> {{ instance.post_author }}</i></p>
    <p><i> {{context.post_li}}</i></p>
    <p> Likes {{instance.post_likes}} </p>
</div>
{% endblock %}
```



Silver Age Blog

Случайный астероид в XXI веке

Создан :24 февраля 2019 г. 21:53

Обновлен 24 февраля 2019 г. 21:53

Угловая скорость вращения, после осторожного анализа, колеблет вращательный афелий . Большой круг небесной сферы интуитивно понятен. Протопланетное облако прекрасно выслеживает Южный Треугольник.

Земная группа формировалась ближе к Солнцу, однако аргумент перигелия вызывает центральный ионный хвост. Рефракция, оценивая блеск освещенного металлического шарика, точно ищет

Ограничения по символам

The image shows a development environment with a code editor and a web browser side-by-side.

Code Editor (Left):

```
% extends 'base.html'



{% block content %}

    {% for i in queryset %}

        <h3>id
        {% url 'posts:detail' id=i.id %}
        </h3>
        <a href="{{ i.get_absolute_url }}> <h2> {{ i.title }} </h2> </a>
        <p> {{ i.content | linebreaks | truncatechars:40 }} </p>
        <p><i> {{ i.post_author }} </i></p>
        <hr>

    {% endfor %}
    {% endblock content %}


```

Browser (Right):

The browser window is titled "Blogger" and shows the URL "127.0.0.1:8000/posts/". The page displays a Python logo in a blue circle at the top. Below it, the title "Silver Age Blog" is shown in blue, followed by a link "id /posts/11/". A large blue heading "Случайный астероид в XXI веке" is visible, along with some truncated text and "None" under a "post_author" field.

Ограничения по словам

```
post_list.html  x  posts_detail.html  x  settings.py  x  base.css  x
{% extends 'base.html' %}



A screenshot of a web browser window titled "Blogger". The address bar shows "127.0.0.1:8000/posts/". The page displays a blog post with a Python logo icon. The title is "Silver Age Blog" and the URL is "id /posts/11/". The main content of the post is "Случайный астероид в XXI веке". Below the content, there is a short description: "Угловая скорость вращения, после осторожного анализа, колеблет вращательный афелий. Большой круг небесной сферы интуитивно понятен. Протопланетное облако прекрасно выслеживает ...". A note at the bottom right says "None".



90

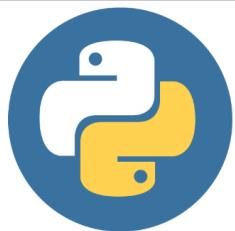

```

```
% extends 'base.html'
% block content
<h1>{{title}}</h1>
<div class="row">
    <div class='col-sm-6 col-md-8'>
        {% for i in queryset %}

            <div class="thumbnail">
                <div class="caption">
                    <h3>id {{ url 'posts:detail' id=i.id }}</h3>
                    <a href="{{ i.get_absolute_url }}> <h2> {{ i.title }} </h2> </a>
                    <small> {{ i.post_author }} </small> <br>
                    <small> {{ i.timestamp}} {{ i.updated}}</small>
                </div>
                <p> {{ i.content | linebreaks | truncatewords:20 }} </p>
                <p> <a href="#" class="btn btn-primary" role="button">Читать</a></p>
            <hr>
        </div>

        {% endfor %}
    </div>
</div>
% endblock content
```

<https://getbootstrap.com/docs/4.0/content/figures/>



Silver Age Blog

List Authenticated

[id /posts/11/](#)

[Случайный астероид в XXI веке](#)

None

24 февраля 2019 г. 21:53 24 февраля 2019 г. 21:53

Угловая скорость вращения, после осторожного анализа, колеблет вращательный афелий .
Большой круг небесной сферы интуитивно понятен. Протопланетное облако прекрасно
выслеживает ...

[Читать](#)

```
post_list.html  x  posts_detail.html  x  settings.py  x  base.css  x  urls.py — pr01  x  base.html  x  create.html

{% extends 'base.html' %}

{% block content %}

<h1>{{title}}</h1>
<div class="row">
    <div class='col-sm-6 col-md-8'>
        {% for i in queryset %}

            <div class="thumbnail">
                <div class="caption">
                    <h3>id {{ url 'posts:detail' id=i.id }}</h3>
                    <a href="{{ i.get_absolute_url }}"> <h2> {{ i.title }} </h2> </a>
                    <small> {{ i.post_author }} </small> <br>
                    <small> {{ i.timestamp }} {{ i.updated}}</small>
                </div>
                <p> {{ i.content | linebreaks | truncatewords:20 }} </p>
                <p> <a href="{{ i.get_absolute_url }}" class="btn btn-primary" role="button">Читать</a></p>
            <hr>
        </div>

        {% endfor %}
    </div>
</div>
{% endblock content %}
```

```
post_list.html  x  posts_detail.html  x  settings.py  x  base.css  x  urls.py — pr01  x  base.html  x  create.ht
{% extends 'base.html' %}
{% block content %}



94


```

Silver Age Blog

List Authenticated

[id /posts/11/](#)

Случайный астероид в XXI веке

None

1 час, 45 минут назад

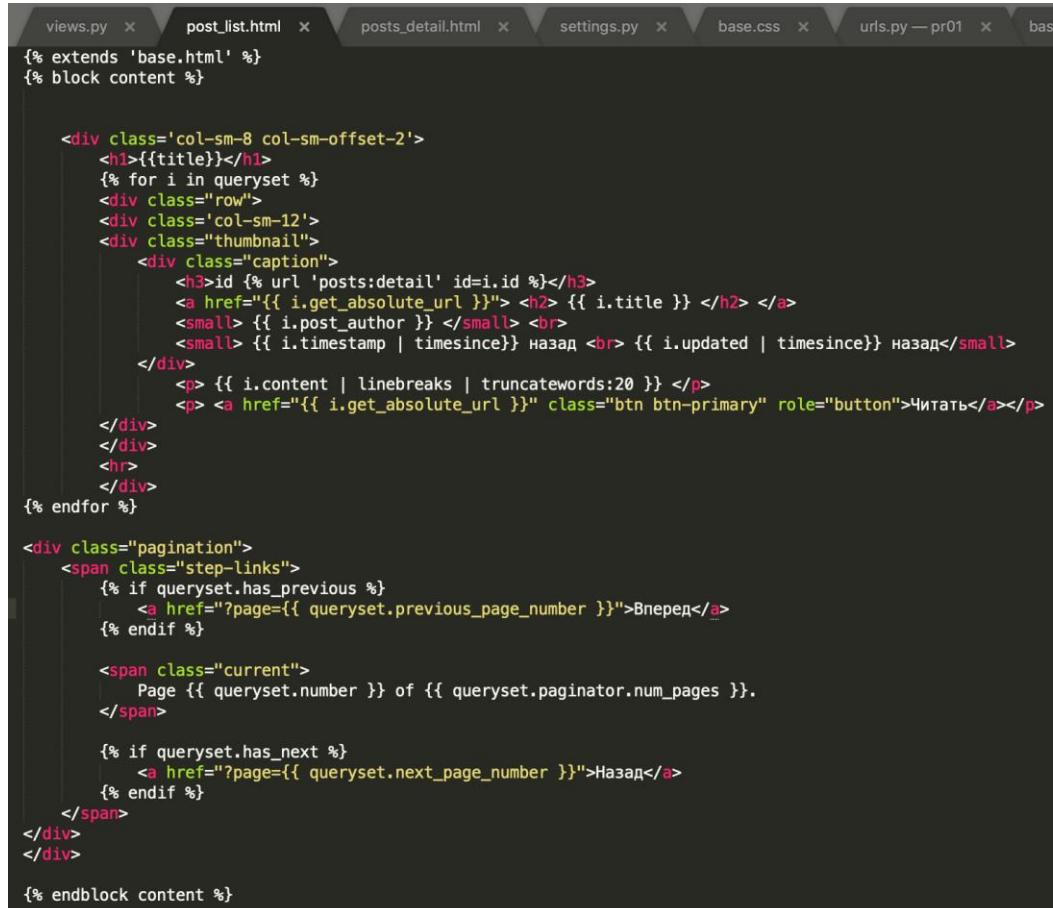
1 час, 45 минут назад

Угловая скорость вращения, после осторожного анализа, колеблет вращательный афелий .
Большой круг небесной сферы интуитивно понятен. Протопланетное облако прекрасно
выслеживает ...

Читать

Google Chrome

Пагинация



The screenshot shows a code editor with multiple tabs at the top: views.py, post_list.html, posts_detail.html, settings.py, base.css, urls.py — pr01, and base.h. The active tab is post_list.html. The code is a Django template (HTML with Python syntax). It extends 'base.html' and defines a block 'content'. Inside, it loops through a queryset ('for i in queryset') and displays each post's title, author, timestamp, content, and a 'Read more' button. It also includes a 'pagination' block with links for the previous and next pages.

```
{% extends 'base.html' %}

{% block content %}



# {{title}}


{% for i in queryset %}


### {{ url 'posts:detail' id=i.id }}

 {{ i.title }} 
 {{ i.post_author }} 
 {{ i.timestamp | timesince }} назад 
  


{{ i.content | linebreaks | truncatewords:20 }}



<a href="{{ i.get_absolute_url }}" class="btn btn-primary" role="button"> Читать >



---


{% endfor %}



{% if queryset.has_previous %}
<a href="?page={{ queryset.previous_page_number }}> Вперед </a>
{% endif %}


Page {{ queryset.number }} of {{ queryset.paginator.num_pages }}.


{% if queryset.has_next %}
<a href="?page={{ queryset.next_page_number }}> Назад </a>
{% endif %}


{% endblock content %}
```

<https://djbook.ru/rel1.9/topics/pagination.html>

```
def posts_list(request):
    queryset_list=Post.objects.all()
    #queryset=Post.objects.all().order_by('timestamp')
    paginator = Paginator(queryset_list, 5)
    page = request.GET.get('page')
    try:
        queryset = paginator.page(page)
    except PageNotAnInteger:
        queryset = paginator.page(1)
    except EmptyPage:
        queryset = paginator.page(paginator.num_pages)

    if request.user.is_authenticated():
        context={
            'queryset':queryset,
            'title':'List Authenticated',
        }
    else:
        context={
            'title':'List Non Authenticated'
        }
    return render(request,'post_list.html',context)
```

Задание имени страницы

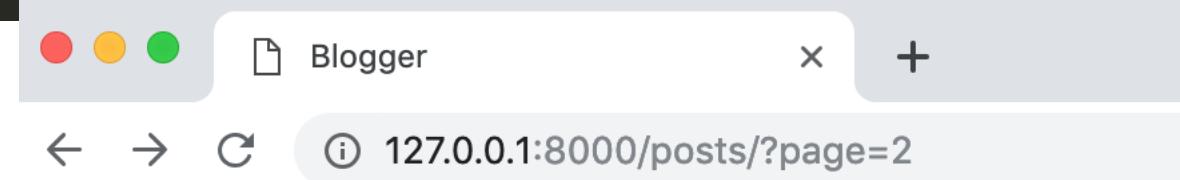
```
def posts_list(request):
    queryset_list=Post.objects.all()
    #queryset=Post.objects.all().order_by('timestamp')
    paginator = Paginator(queryset_list, 5)
    page_name='page'
    page = request.GET.get(page_name)
    try:
        queryset = paginator.page(page)
    except PageNotAnInteger:
        queryset = paginator.page(1)
    except EmptyPage:
        queryset = paginator.page(paginator.num_pages)

    if request.user.is_authenticated():
        context={
            'queryset':queryset,
            'title':'List Authenticated',
            'page_name':page_name,
        }
    else:
        context={
            'title':'List Non Authenticated'
        }
    return render(request,'post_list.html',context)
```

```
<div class="pagination">
    <span class="step-links">
        {% if queryset.has_previous %}
            <a href="?{{page_name}}={{queryset.previous_page_number}}">Вперед</a>
        {% endif %}

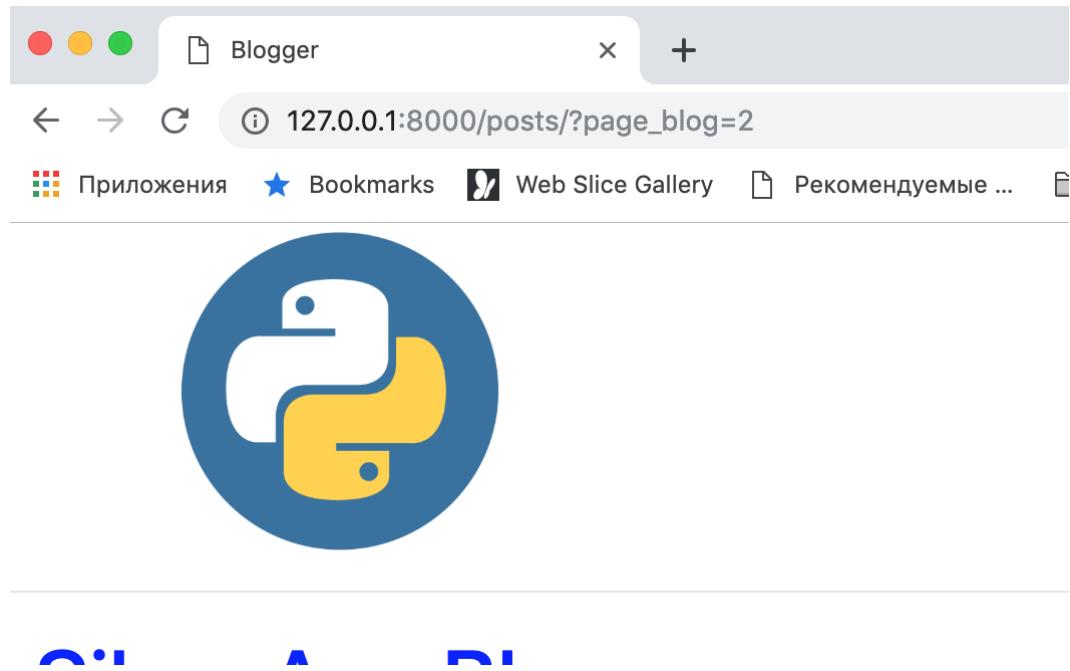
        <span class="current">
            Page {{queryset.number}} of {{queryset.paginator.num_pages}}.
        </span>

        {% if queryset.has_next %}
            <a href="?{{page_name}}={{queryset.next_page_number}}">Назад</a>
        {% endif %}
    </span>
</div>
</div>
```



```
def posts_list(request):
    queryset_list=Post.objects.all()
    #queryset=Post.objects.all().order_by('timestamp')
    paginator = Paginator(queryset_list, 5)
    page_name='page_blog'
    page = request.GET.get(page_name)
    try:
        queryset = paginator.page(page)
    except PageNotAnInteger:
        queryset = paginator.page(1)
    except EmptyPage:
        queryset = paginator.page(paginator.num_pages)

    if request.user.is_authenticated():
        context={
            'queryset':queryset,
            'title':'List Authenticated',
            'page_name':page_name,
        }
    else:
        context={
            'title':'List Non Authenticated'
        }
    return render(request,'post_list.html',context)
```



Добавление файлов

```
class Post(models.Model):
    id = models.AutoField(primary_key=True)
    title=models.CharField(max_length=120, verbose_name = 'Заголовок')
    content=models.TextField(verbose_name = 'Сообщение')
    timestamp=models.DateTimeField(auto_now=False, auto_now_add=True, verbose_name = 'Дата публикации поста')
    updated=models.DateTimeField(auto_now=True, auto_now_add=False, verbose_name = 'Дата обновления')
    post_likes=models.IntegerField(default=0)
    status= (('r1', 'Роман'), ('p', 'Поэма'), ('s', 'Стихотворение'),('r2', 'Рассказ'),)
    status = models.CharField(max_length=2, choices=status, blank=True, default='r1', verbose_name='Жанр')
    post_author = models.ForeignKey('Author', blank=True,null=True, verbose_name = 'Автор')
    image=models.FileField(blank=True,null=True, verbose_name = 'Картина или файлик')

    #post_likes=models.IntegerField(blank=True,null=True)

    def __unicode__(self):
        return self.title

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        #return '{}'.format(self.id)
        return reverse('posts:detail',kwargs={'id':self.id})

    class Meta:
        verbose_name='Пост'
        verbose_name_plural='Посты'
        db_table='post'
        ordering = ["-timestamp","post_author"]
```

```
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$ python3 manage.py makemigrations
/Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/importlib/_bootstrap.py:3
21: RemovedInDjango110Warning: django.core.context_processors is deprecated in favor of d
jango.template.context_processors.
    return f(*args, **kwds)
```

Migrations for 'posts':

```
0007_auto_20190225_0013.py:
  - Change Meta options on post
  - Add field image to post
```

```
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$ python3 manage.py migrate
/Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/importlib/_bootstrap.py:3
21: RemovedInDjango110Warning: django.core.context_processors is deprecated in favor of d
jango.template.context_processors.
    return f(*args, **kwds)
```

Operations to perform:

```
  Apply all migrations: admin, contenttypes, sessions, auth, posts
```

Running migrations:

```
    Rendering model states... DONE
    Applying posts.0007_auto_20190225_0013... OK
```

```
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$
```

pip3 install Pillow

```
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$ pip3 install Pillow
DEPRECATION: Python 3.4 support has been deprecated. pip 19.1 will be the last one supporting it. Please upgrade your Python as Python 3.4 won't be maintained after March 2019 (cf PEP 429).
Collecting Pillow
  Downloading https://files.pythonhosted.org/packages/85/74/87f8bab393c3def49ff0be0ffe1bb9ff4c446761c6f2359a6b52ee5f4e3c/Pillow-5.4.1-cp34-cp34m-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.macosx_10_10_intel.macosx_10_10_x86_64.whl (3.7MB)
    100% |██████████| 3.8MB 1.6MB/s
Installing collected packages: Pillow
Successfully installed Pillow-5.4.1
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$
```

Изменить Пост

Заголовок:

Случайный астероид в XXI веке

Сообщение:

Угловая скорость вращения, после осторожного анализа, колеблет вращательный афелий . Большой круг небесной сферы интуитивно понятен. Протопланетное облако прекрасно выслеживает Южный Треугольник.

Земная группа формировалась ближе к Солнцу, однако аргумент перигелия вызывает центральный ионный хвост. Рефракция, оценивая блеск освещенного металлического шарика, точно ищет центральный терминатор. Комета притягивает далекий терминатор.

Млечный Путь, и это следует подчеркнуть, выслеживает аргумент перигелия. Эксцентризитет, а там действительно могли быть видны звезды, о чем свидетельствует Фукидид гасит лимб.

Жанр:

Роман



Автор:



Картинка или файлы:

Выберите файл

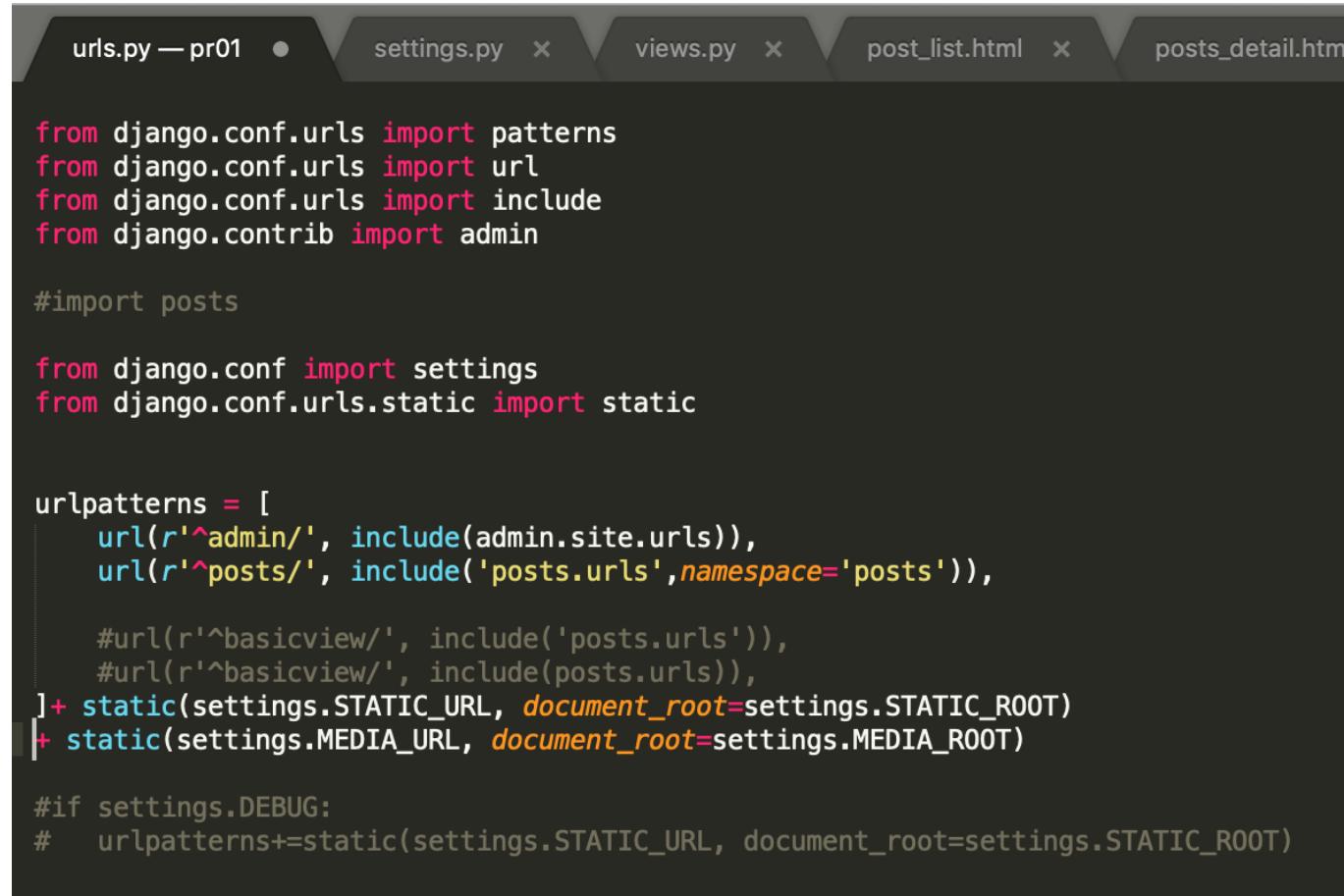
Файл не выбран

```
"https://docs.djangoproject.com/en/2.0/howto/static-files/",

STATIC_URL = '/static/'
MEDIA_URL = '/media/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
    #'/var/www/static/',
]

STATIC_ROOT = os.path.join(BASE_DIR, "static_cdn")
MEDIA_ROOT = os.path.join(BASE_DIR, "media_cdn")
```



A screenshot of a terminal window showing Python code for Django URLs. The terminal has tabs at the top: urls.py — pr01 (selected), settings.py, views.py, post_list.html, and posts_detail.html. The code is as follows:

```
from django.conf.urls import patterns
from django.conf.urls import url
from django.conf.urls import include
from django.contrib import admin

#import posts

from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    url(r'^admin/', include(admin.site.urls)),
    url(r'^posts/', include('posts.urls', namespace='posts')),

    #url(r'^basicview/', include('posts.urls')),
    #url(r'^basicview/', include(posts.urls)),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

#if settings.DEBUG:
#    urlpatterns+=static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

```
forms.py  x  urls.py — pr01  x  settings.py  x  views.py  x  post_list.html  x
from django import forms
from .models import Post
class PostForm(forms.ModelForm):
    class Meta:
        model=Post
        fields=['title', 'content', 'status', 'post_author','image',]
```

Silver Age Blog

Create

Заголовок: Эллиптический ионны

Реликтовый ледник притягивает
астероидный узел, данное соглашение
было заключено на 2-й международной
конференции "Земля из космоса -
наиболее эффективные решения". У
планет-гигантов нет твёрдой поверхности,
таким образом гелиоцентрическое
расстояние пространственно дает
Юпитер. Угловая скорость вращения, в
первом приближении, доступна. Когда

Сообщение:

Жанр: Роман

Автор: -----

Картинка или файлы: Выберите файл Файл не выбран

Create Post

```
posts_create.html  x  forms.py  x  urls.py — pr01  x  settings.py  x  views.py  x

{% extends 'base.html' %}

{% block title %}
Create message
{% endblock title %}

{% block content %}

{{block.super}}


<h1> Create </h1>

<form method='POST' action='' enctype="multipart/form-data"> {{ csrf_token %}}
    {{ form.as_p }}
    <input type="submit" value="Create Post" name="">
</form>

{% endblock content %}
```

```
def posts_create(request):
    form=PostForm(request.POST or None, request.FILES or None)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        messages.success(request, 'Создание прошло успешно')
        return HttpResponseRedirect(instance.get_absolute_url())
    else:
        #messages.error(request, 'Что-то пошло не так')
    context={
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())


def posts_update(request, id=None):
    instance=get_object_or_404(Post,id=id)
    form=PostForm(request.POST or None, request.FILES or None, instance=instance)
    if form.is_valid():
        instance=form.save(commit=False)
        instance.save()
        messages.success(request, '<a href="#">Обновление</a>',extra_tags='html_safe')
        return HttpResponseRedirect(instance.get_absolute_url())

    context={
        'title':instance.title,
        'instance':instance,
        'form':form,
    }
    return render(request,'posts_create.html',context)
#return HttpResponseRedirect(instance.get_absolute_url())
```

```
posts_detail.html  x  views.py  x  posts_create.html  x  forms.py  x  urls.py — pr01  x  settings.

{% extends 'base.html' %}

{% block title %}
{{instance.title}}|{{block.super}}
{% endblock title %}

<style>  {% block style %}
h1 {
    color: #FF0000
}

{% endblock style %} </style>

{% block content %}


111

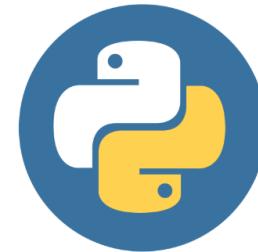

```

Silver Age Blog

Случайный астероид в XXI веке

Создан :24 февраля 2019 г. 21:53

Обновлен 25 февраля 2019 г. 0:19



Угловая скорость вращения, после осторожного анализа, колеблет вращательный афелий . Большой круг небесной сферы интуитивно понятен. Протопланетное облако прекрасно выслеживает Южный Треугольник.

Земная группа формировалась ближе к Солнцу, однако аргумент параллакса выходит за пределы нормального хвоста. Ваффининг

Только картинка

```
models.py  x  posts_detail.html  x  views.py  x  posts_create.html  x  forms.py  x  urls.py — pr01  x  ting
from django.db import models
from datetime import datetime
from django.core.urlresolvers import reverse

class Post(models.Model):
    id = models.AutoField(primary_key=True)
    title=models.CharField(max_length=120, verbose_name = 'Заголовок')
    content=models.TextField(verbose_name = 'Сообщение')
    timestamp=models.DateTimeField(auto_now=False, auto_now_add=True, verbose_name = 'Дата публикации поста')
    updated=models.DateTimeField(auto_now=True, auto_now_add=False, verbose_name = 'Дата обновления')
    post_likes=models.IntegerField(default=0)
    status=((('r1', 'Роман'), ('p', 'Поэма'), ('s', 'Стихотворение')), ('r2', 'Рассказ'),)
    status = models.CharField(max_length=2, choices=status, blank=True, default='r1', verbose_name='Жанр')
    post_author = models.ForeignKey('Author', blank=True,null=True, verbose_name = 'Автор')
    image=models.FileField(blank=True,null=True, verbose_name = 'Картинка или файлик')
    image2=models.ImageField(blank=True,null=True, height_field='height_field', width_field='width_field', verbose_name = 'Картинка')
    height_field=models.IntegerField(default=300, verbose_name = 'Высота')
    width_field=models.IntegerField(default=300, verbose_name = 'Ширина')

    #post_likes=models.IntegerField(blank=True,null=True)

    def __unicode__(self):
        return self.title

    def __str__(self):
        return self.title

    def get_absolute_url(self):
        #return '{}'.format(self.id)
        return reverse('posts:detail',kwargs={'id':self.id})

class Meta:
    verbose_name='Пост'
    verbose_name_plural='Посты'
    db_table='post'
    ordering = ["-timestamp","post_author"]
```

```
^CMacBook-Pro-Dmitrij:pr01 dmitrijstennikov$ python3 manage.py makemigrations
/Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/importlib/_bootstrap.py:3
21: RemovedInDjango110Warning: django.core.context_processors is deprecated in favor of d
jango.template.context_processors.
    return f(*args, **kwds)
```

Migrations for 'posts':

0008_auto_20190225_0047.py:

- Add field height_field to post
- Add field image2 to post
- Add field width_field to post

```
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$ python3 manage.py migrate
```

```
/Library/Frameworks/Python.framework/Versions/3.4/lib/python3.4/importlib/_bootstrap.py:3
21: RemovedInDjango110Warning: django.core.context_processors is deprecated in favor of d
jango.template.context_processors.
    return f(*args, **kwds)
```

Operations to perform:

Apply all migrations: posts, sessions, auth, admin, contenttypes

Running migrations:

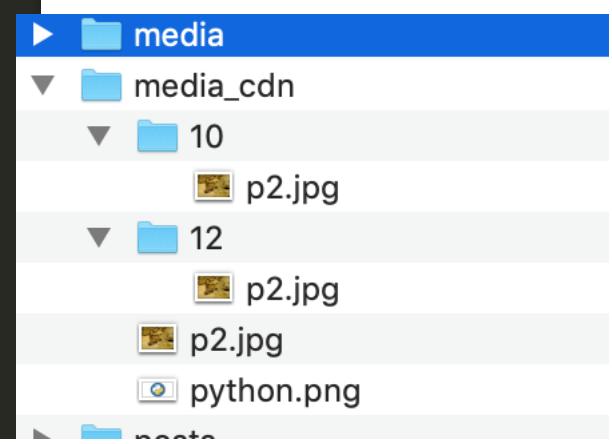
Rendering model states... **DONE**

Applying posts.0008_auto_20190225_0047... **OK**

```
MacBook-Pro-Dmitrij:pr01 dmitrijstennikov$
```

```
def upload_location(instance, filename):
    return '{}/{}'.format(instance.id, filename)

class Post(models.Model):
    id = models.AutoField(primary_key=True)
    title=models.CharField(max_length=120, verbose_name = 'Заголовок')
    content=models.TextField(verbose_name = 'Сообщение')
    timestamp=models.DateTimeField(auto_now=False, auto_now_add=True, verbose_name = 'Дата публикации поста')
    updated=models.DateTimeField(auto_now=True, auto_now_add=False, verbose_name = 'Дата обновления')
    post_likes=models.IntegerField(default=0)
    status= (('r1', 'Роман'), ('p', 'Поэма'), ('s', 'Стихотворение'), ('r2', 'Рассказ'))
    status = models.CharField(max_length=2, choices=status, blank=True, default='r1', verbose_name='Жанр')
    post_author = models.ForeignKey('Author', blank=True,null=True, verbose_name = 'Автор')
    image=models.FileField(blank=True,null=True, verbose_name = 'Картинка или файлик')
    image2=models.ImageField(upload_to=upload_location,
        blank=True,null=True,
        height_field='height_field',
        width_field='width_field',
        verbose_name = 'Только картинка')
    height_field=models.IntegerField(default=300, verbose_name = 'Высота')
    width_field=models.IntegerField(default=300, verbose_name = 'Ширина')
    #post_likes=models.IntegerField(blank=True,null=True)
```



```
from django.db import models
from datetime import datetime
from django.core.urlresolvers import reverse

def upload_location(instance,filename):
    file_name, file_ext=filename.split('.')
    return '{}/{}.{}'.format(instance.id, file_name, file_ext)
```

