

Урок 6

Декораторы

```
>>> def add(x, y):
```

```
...     return x + y
```

```
>>> def sub(x, y):
```

```
...     return x - y
```

```
>>> def apply(func, x, y): # 1
```

```
...     return func(x, y) # 2
```

```
>>> apply(add, 2, 1) # 3
```

```
3
```

```
>>> apply(sub, 2, 1)
```

Декораторы

```
>>> def outer():
```

```
...     def inner():
```

```
...         print "Inside inner"
```

```
...     return inner # 1
```

```
...
```

```
>>> foo = outer() #2
```

```
>>> foo # doctest:+ELLIPSIS
```

```
<function inner at 0x...>
```

```
>>> foo()
```

```
Inside inner
```

Декораторы.

```
>>> def outer(some_func):  
...     def inner():  
...         print "before some_func"  
...         ret = some_func() # 1  
...         return ret + 1  
...     return inner  
  
>>> def foo():  
...     return 1  
  
>>> decorated = outer(foo) # 2  
  
>>> decorated()
```

Знакомство с редактором VIM.

Просмотр категории

Добавим вьюхи

-----blog/view.py-----

```
from django.views.generic import ListView, DetailView
```

```
from blog.models import BlogCategory
```

```
class CategoryDetailView(DetailView):
```

```
    model = BlogCategory
```

```
    template_name = 'blogcategory_detail.html'
```

Создадим шаблон для отображения

blog/templates/blogcategory_detail.html

Пропишем роутинг в urls.py

```
from blog.views import CategoryDetailView
```

```
url(r'^blog-category/(?P<pk>\d+)/$', CategoryDetailView.as_view()),
```

Шаблон

```
{% extends "base.html" %}
{% block base_content %}
<h2>{{ blogcategory }}</h2>
    {% for i in blogcategory.blogtopic_set.all %}
        <div class="col-sm-6">
            <h3> {{ i.title }}</h3>
            <p> {{ i.content|truncatewords_html:20 }}</p>
            <p><a class="btn btn-default" href="#" role="button">Подробнее &raquo;</a></p>
        </div><!--/span-->
    {% endfor %}
{% endblock %}
```

Ссылка на пост

Создадим ссылку на пост блога в модели

-----blog/models.py

```
class BlogTopic(models.Model):
```

```
.....
```

```
    def get_absolute_url(self):
```

```
        return "/blog-topic/%i/" % self.id
```

Создадим ссылку в шаблоне blog/templates/blogcategory_detail.html

```
<p><a class="btn btn-default" href="{ i.get_absolute_url }" role="button">Подробнее &raquo;</a></p>
```


Подробная информация

Создадим вид в `blog/view.py`

```
from blog.models import BlogTopic
class TopicDetailView(DetailView):
    model = BlogTopic
    template_name = 'blogtopic_detail.html'
```

Пропишем роутинг в `urls.py`

```
from blog.views import TopicDetailView

url(r'^blog-topic/(?P<pk>\d+)/$', TopicDetailView.as_view()),
```