

Демонстрация ЛР:  
Гапанюк. Ю.Е.

**Отчет по лабораторной работе № 5 по курсу  
РИП**

**"Шаблонизация"**

ИСПОЛНИТЕЛЬ:

студент группы **ИУ5-54**

**Наседкин И.А.**

## Задание ЛР

В этой ЛР вы создадите Django-проект, покажете пользователю статичную страницу, познакомитесь с конструкциями шаблонизаторов: переменные, теги, наследование шаблонов.

- Создать проект
- Реализовать view, в которых генерируются html-страницы
- В шаблонах должны быть использованы рассмотренные конструкции: переменные, вложенные значения, циклы, условия
- Все шаблоны должны расширять базовый шаблон
- Для элементов списка использовать тег include
- По нажатии на элемент списка должна открываться страница информации об элементе
- Для верстки необходимо использовать Bootstrap

## Текст программы

### views.py

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

# Create your views here.
def function_view(request):
    return HttpResponseRedirect('response from function view')

class ExampleClassBased(View):
    def get(self, request):
        return HttpResponseRedirect('response from class based view')

class ExampleView(View):
    def get(self, request):
        return render(request, 'myfile.html', {'dict': {'inner': 'a'}})

class OrdersView(View):
    def get(self, request):
        data = {
            'orders': [
                {'title': 'Первый заказ', 'id': 1},
                {'title': 'Второй заказ', 'id': 2},
                {'title': 'Третий заказ', 'id': 3}
            ]
        }
        return render(request, 'orders.html', data)

class OrderView(View):
    def get(self, request, id):
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', data)
```

### urls.py

```
"""mysite URL Configuration
```

```
The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/1.9/topics/http/urls/
Examples:
```

Function views

1. Add an import: `from my_app import views`
2. Add a URL to urlpatterns: `url(r'^$', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to urlpatterns: `url(r'^$', Home.as_view(), name='home')`

Including another URLconf

1. Add an import: `from blog import urls as blog_urls`
2. Import the `include()` function: `from django.conf.urls import url, include`
3. Add a URL to urlpatterns: `url(r'^blog/', include(blog_urls))`

"""

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

```
from polls.views import ExampleClassBased, function_view, ExampleView, OrderView,
OrdersView
```

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^function_view/', function_view),
    url(r'^class_based_view/', ExampleClassBased.as_view()),
    url(r'^orders_view/', OrdersView.as_view()),
    url(r'^order/(?P<id>\d+)', OrderView.as_view(), name='order_url'),
]
```

## myfile.html

```
{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.css'
%}">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    <nav class="navbar navbar-light" style="background-color: #e3f2fd;">

    <div class="container-fluid">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand"><em>mysite</em></a>
        </div>

        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav">
                <li><a href="/orders_view/">Myfile.html</a></li>
                {% block header %}{% endblock %}
            </ul>
        </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>

<div>
    {% block body %}Переопределяется в наследниках{% endblock %}
```

```
</div>
</body>
</html>
```

## inc.html

```
{{ verb }}
```

## order.html

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

{% block title %}Заказ №{{ order.id }}{% endblock %}

{% block body %}Страница заказа №{{ order.id }}{% endblock %}
```

## order\_element.html

```
<li><a href="{% url 'order_url' id=element.id %}">{{ element.title }}</a></li>
```

## orders.html

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

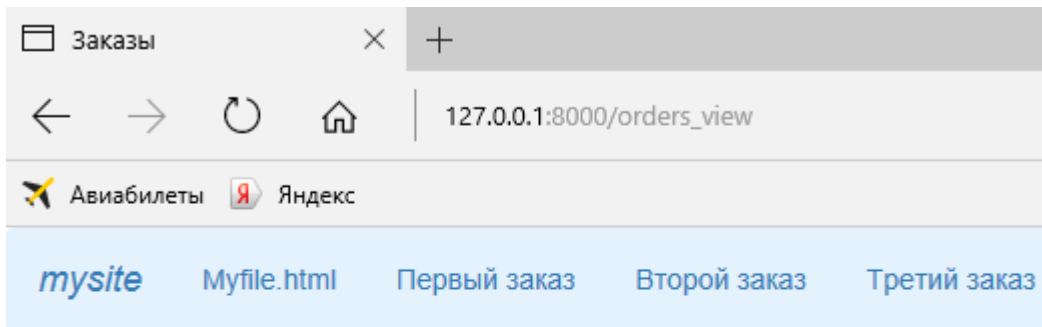
{% block title %} Заказы{% endblock %}

{% block body %}
    <ul class="list-group">
        {% for order in orders %}
            <li class="list-group-item"><a href="{% url 'order_url' order.id
%}">{{ order.title }}</a></li>
            {% empty %}
                <li class="list-group-item disabled">пустой список</li>
            {% endfor %}
        </ul>
    {% endblock %}

{%block header %}
{% for order in orders %}
    {%include 'order_element.html' with element=order%}
    {% endfor %}
{% endblock %}
```

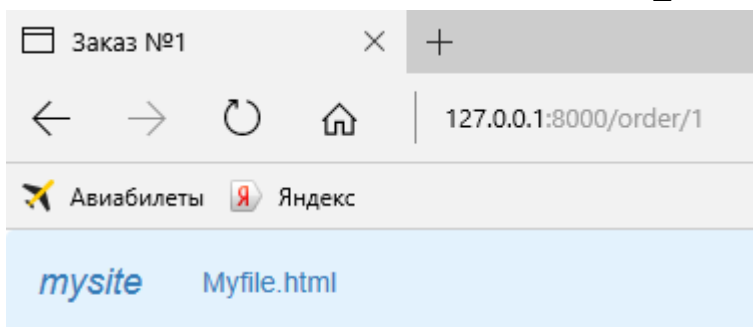
## Результаты работы программы

1



<a href="#">Первый заказ</a>
<a href="#">Второй заказ</a>
<a href="#">Третий заказ</a>

2



Страница заказа №1