

Факультет «Информатика и системы управления»  
Кафедра «Системы обработки информации и управления»



Лабораторные работы по курсу:  
**«Разработка Интернет Приложений»**

**ЛР5. Python. Шаблонизация**

Исполнитель:

Студент группы РТ5-51

Умряев Д.Т.

Преподаватель:

Гапанюк Ю. Е.

«    »



## Задание и порядок выполнения

В этой ЛР необходимо создать Django-проект, показать пользователю статичную страницу, познакомиться 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

## Исходный код:

### **settings.py**

```
"""
```

```
Django settings for lab5 project.
```

```
Generated by 'django-admin startproject' using Django 1.11.6.
```

```
For more information on this file, see
```

```
https://docs.djangoproject.com/en/1.11/topics/settings/
```

```
For the full list of settings and their values, see
```

```
https://docs.djangoproject.com/en/1.11/ref/settings/
```

```
"""
```

```
import os
```

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
```

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
```

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
```

```
SECRET_KEY = '49!#b03k-$776(y!yk)1i== $de40779llydo(okzdn3($ca!c7'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

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

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

```
ROOT_URLCONF = 'lab5.urls'
```

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [os.path.join(BASE_DIR, 'templates')]  
    },  
    {  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    },  
]
```

```
WSGI_APPLICATION = 'lab5.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases
```

```
DATABASES = {  
    'default': {
```

```
    'ENGINE': 'django.db.backends.sqlite3',
    'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
}
}
```

# Password validation

# <https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators>

```
AUTH_PASSWORD_VALIDATORS = [
    {
        '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/1.11/topics/i18n/>

```
LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'
```

```
USE_I18N = True
```

```
USE_L10N = True
```

```
USE_TZ = True
```

# Static files (CSS, JavaScript, Images)

# <https://docs.djangoproject.com/en/1.11/howto/static-files/>

```
STATIC_URL = '/static/'
```

**lab5/urls.py**

```
"""lab5 URL Configuration
```

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/1.11/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. Import the `include()` function: `from django.conf.urls import url, include`
2. Add a URL to `urlpatterns`: `url(r'^blog/', include('blog.urls'))`

"""

```
from django.conf.urls import url, include
from django.contrib import admin
from my_app.views import OrdersView, main, prog_lang
from django.conf.urls.static import static
from django.conf import settings
```

```
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^orders/', include('my_app.urls')),
    url(r'^orders/', OrdersView.as_view(), name='orders_url'),
    url(r'^main/', main, name='main_url'),
    url(r'^(?P<id>\d+)', prog_lang, name='prog_lang_url'),
]
```

```
if settings.DEBUG:
```

```
    urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_URL)
```

### **my\_app/urls.py**

```
from django.conf.urls import url
from my_app.views import OrderView
```

```
urlpatterns = [
    url(r'^(?P<id>\d+)', OrderView.as_view(), name='order_url'),
]
```

### **views.py**

```
from django.shortcuts import render
from django.views import View
from datetime import datetime
# Create your views here.
```

```
class OrdersView(View):
```

```
def get(self, request):
    variable = 'Django'
    today_date = datetime.now()
    data = {
        'orders': [
            {'title': 'Первый заказ', 'id': 1},
            {'title': 'Второй заказ', 'id': 2},
            {'title': 'Третий заказ', 'id': 3}
        ]
    }
    return render(request, 'orders.html', locals())
```

```
class OrderView(View):
    def get(self, request, id):
        variable = 'Django'
        today_date = datetime.now()
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', locals())
```

```
def main(request):
    return render(request, 'main.html', locals())
```

```
def prog_lang(request, id):
    name = ['C++', 'Python', 'Java']
    cpp_info = 'C++ — компилируемый, статически типизированный язык программирования
общего назначения. Синтаксис C++ унаследован от языка C. Одним из принципов разработки
было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле
надмножеством C; множество программ, которые могут одинаково успешно транслироваться как
компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные
программы на C.'
```

java\_info = 'Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.'

```
python_info = 'Python — высокоуровневый язык программирования общего назначения,
ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис
ядра Python минималистичен. В то же время стандартная библиотека включает большой объём
полезных функций.'
```

```
info = [cpp_info, java_info, python_info]
data1 = {'lang': {'id': id}}
data2 = {'langs': [{'id': '1', 'lang_name': 'C++', 'info': cpp_info},
```

```
        {'id': '2', 'lang_name': 'Java', 'info': java_info},
        {'id': '3', 'lang_name': 'Python', 'info': python_info}}]
return render(request, 'prog_lang.html', locals())
```

### **main.css**

```
@import url('https://fonts.googleapis.com/css?family=Lato|Play');
```

```
body {
    background-color: #f2f2f2;
    font-family: Lato, Play;
    font-weight: 700;
    font-size: 25px;
}
```

```
h1, h2, h3, h4, h5 ,h6 {
    font-family: Play;
    color: #999999;
}
```

```
p {
    font-family: Play;
    color: #999999;
    text-align: left;
    line-height: 28px;
    margin-bottom: 25px;
}
```

```
a {
    color: darkgray;
}
```

```
a:hover, a:focus {
    color: #ffffff;
    text-decoration: none;
}
```

```
li {
    text-align: left;
}
```

```
.navbar {
    background-color: #444444;
    padding-top: 15px;
    padding-bottom: 15px;
}
```

```
.navbar-brand {
```

```

    font-weight: 600;
    font-size: 20px;
}

#header1 {
    background: url(../../static/images/bg1.jpg) no-repeat;
    text-align: center;
    margin-top: 80px;
    padding-top: 15px;
    min-height: 1080px;
    background-size: 100%;
}

.container {
    width: 90%;
    color: #999999;
}

```

## base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    {% load static %}
    <link rel="stylesheet" href="{ % static 'Bootstrap/css/bootstrap.css' % }">
    <link rel="stylesheet" href="{ % static 'Bootstrap/css/font-awesome.min.css' % }">
    <link rel="stylesheet" href="{ % static 'Bootstrap/css/main.css' % }">
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="#">MYA<i class="fa fa-rub" aria-hidden="true"></i><i
class="fa fa-rub" aria-hidden="true"></i></a>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav navbar-right">
                    <li><a href="{ % url 'main_url' % }">Home</a></li>
                    <li><a href="{ % url 'orders_url' % }">Task</a></li>
                </ul>

```



```

        </div>
    </div>
</div>
{% block body1 %}{% endblock %}
</body>
</html>

```

## main.html

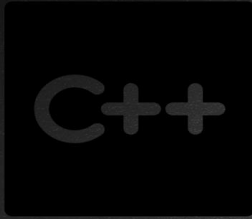
```

{% extends 'base.html' %}

{% block body1 %}
    <div id="header1">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h1>Programming Languages</h1>
                </div>
            </div>
        </div>
        <br><br>
        <div class="container">
            <div class="row centered">
                {% load static %}
                <div class="col-lg-4">
                    <a href="{% url 'prog_lang_url' 1 %}"></a>
                </div>
                <div class="col-lg-4">
                    <a href="{% url 'prog_lang_url' 2 %}"></a>
                </div>
                <div class="col-lg-4">
                    <a href="{% url 'prog_lang_url' 3 %}"></a>
                </div>
            </div>
        </div>
    </div>
{% endblock %}

```

## Programming Languages

**prog\_lang.html**

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

{% block body1 %}
<div id="header1">
  <div class="container">
    <div class="row centered">
      <div class="col-lg-8 col-lg-offset-2">
        <h1>
          {% for lng in data2.langs %}
            {% if lng.id == data1.lang.id %}
              {{ lng.lang_name }}
            {% endif %}
          {% empty %}
            empty list :(
          {% endfor %}
        </h1>
      </div>
    </div>
  </div>
</div>
<br><br>
<div class="col-lg-8 col-lg-offset-2">
  <p>
    {% for lng in data2.langs %}
      {% if lng.id == data1.lang.id %}
        {{ lng.info }}
      {% endif %}
    {% empty %}
      empty list :(
```

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

MYAPP

HomeTask

C++

C++ — компилируемый, статически типизированный язык программирования общего назначения. Синтаксис C++ унаследован от языка C. Одним из принципов разработки было сохранение совместимости с C. Тем не менее, C++ не является в строгом смысле надмножеством C; множество программ, которые могут одинаково успешно транслироваться как компиляторами C, так и компиляторами C++, довольно велико, но не включает все возможные программы на C.

MYAPP

HomeTask

Java

Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.

## Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объем полезных функций.

### orders.html

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

{% block body1 %}
<div id="header1">
  <div class="container">
    <div class="row centered">
      <div class="col-lg-8 col-lg-offset-2">
        <h1>Заказы</h1>
      </div>
    </div>
  </div>
  <div class="col-lg-8 col-lg-offset-2">
    <p>
      <ul>
        {% for order in data.orders %}
          {% include 'order_element.html' with element=order %}
        {% empty %}
          empty list :(
        {% endfor %}
      </ul>
    </p>
  </div>
</div>
{% endblock %}
```

## Заказы

- Первый заказ
- Второй заказ
- Третий заказ

**order.html**

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

{% block body1 %}
  <div id="header1">
    <div class="container">
      <div class="row centered">
        <div class="col-lg-8 col-lg-offset-2">
          <h1>Заказ №{{ data.order.id }}</h1>
        </div>
      </div>
    </div>
    <div class="col-lg-8 col-lg-offset-2">
      <p>Страница заказа №{{ data.order.id }}</p>
    </div>
  </div>
{% endblock %}
```



## Заказ №1

Страница заказа №1

## Заказ №2

Страница заказа №2

Страница заказа №3

Заказ №3

**order\_element.html**

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