# Шаблоны

| **№ урока:** | 4 | **Курс:** | Django Starter |
|---|---|---|---|

| **Средства обучения:** | Персональный компьютер с установленными:<br>Python 3.8<br>Django 3.0 |
|---|---|

## Обзор, цель и назначение урока

Целью данного урока является ознакомление с основами шаблонов в Django. Шаблоны позволяют генерировать один и тот же текст с вариациями в зависимости от данных, в частности отрисовывать один и тот же стиль веб-страницы и т.д.

## Изучив материал данного занятия, учащийся сможет:

- Создавать и использовать шаблоны, а также правильно пользоваться их элементами: тегами, фильтрами, переменными и комментариями.
- Понимать, как взаимодействуют между собой представления и модели.

## Содержание урока

1. Шаблонизатор Django
2. Способы загрузки шаблонов
3. Синтаксис шаблонов: переменные, теги, фильтры и комментарии.
4. Шаблонизатор Jinja: что это и зачем он может быть нужен

## Резюме

- Шаблоны в Django используются для динамической генерации HTML. Они содержат статичный текст и динамические данные в специальном синтаксисе.
- Django содержит в себе поддержку двух backend для обработки шаблонов: **DTL (Django Template Language)** и **Jinja2**.
- Выбранный backend и путь к директории с шаблонами указываются в **settings.py** с помощью списка словарей **TEMPLATES**, где каждый словарь отвечает за настройку одного backend-а.
- Возможные настройки следующие:
    - BACKEND – Python путь к выбранному backend для обработки шаблонов,
    - DIRS – список каталогов, в которых будут искаться шаблоны,
    - APP_DIRS – искать ли шаблоны в установленных приложениях,
    - OPTIONS – содержит настройки, специфичные для выбранного backend-а.
- Загрузка шаблонов осуществляется двумя функциями**: get_template** и **select_template**. Обе находятся в библиотеке django.template.loader.
    - get_template загружает 1 шаблон,
    - select_template позволяет указать список шаблонов, из которых загружается первый доступный.
- Результатом успешной загрузки шаблона одной из этих функций является объект Template, у которого есть метод render(context=None, request=None), который отображает найденный шаблон с указанным context. Контекст используется для задания значений переменных в шаблоне. request – это объект класса django.http.HttpResponse, доступ к которому будет предоставлен в шаблоне.



CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- Для упрощения загрузки у объекта Template есть также метод render_to_string, который объединяет в себе get_template и select_template, и сразу вызывает render.
- Синтаксис вызова render_to_string(template_name, context=None, request=None, using=None).
  Если template_name – str, то вызывается get_template, если это – list, то вызывается select_template.
  При этом context и request такие же, как у метода Template.render.
  Атрибут using позволяет задать имя бэкенда, который будет использоваться для обработки шаблонов.
- **Шаблон Django** – это просто текстовый файл, или строка Python, которые следуют языку шаблонов Django. Определенные конструкции распознаются и интерпретируются шаблонизатором.
- Шаблон рендерится с контекстом. Рендеринг заменяет переменные на их значения, которые ищутся в контексте, и выполняет теги. Все остальное выводится как есть.
- Синтаксис шаблонов для работы с переменными использует 4 конструкции: переменные, теги, фильтры и комментарии.
- Переменные выводят значения из контекста, который является словарем. В шаблоне переменные выделяются **{{** и **}}**.
- Обращение к ключам словаря, атрибутам объектов и элементам списка выполняется через точку. Если значением переменной является вызываемый объект, шаблонизатор вызовет его без аргументов и подставит результат.
- Теги позволяют добавлять произвольную логику в шаблон. Например, теги могут выводить текст, добавлять логические операторы, такие как "if" или "for", получать содержимое из базы данных, или предоставлять доступ к другим тегам.
  Теги выделяются **{%** и **%}**. Большинство тегов принимают аргументы. Некоторые теги требуют определенный закрывающий тег, т.е. являются парными.
- Основные теги в Джанго:
  - {% if %} {% endif %} – отображение с условием. Основные особенности: нельзя использовать скобочки и вложенные сравнения a > b > c.
  - {% for obj in list %} … {% endfor %} – несколько раз выполняет содержимое, подставляя вместо obj элементы list. Как обычный цикл. Можно посередине добавить тег {% empty %}, содержимое после него будет отображаться в случае пустого list.
  - {% cycle 'odd' 'even' %} – возвращает свои аргументы при каждом вызове. При первом будет вызван первый, при втором – второй и т.д., после последнего опять будет первый.
  - {% csrf_token %} – тег, используемый для **csrf-**защиты. Подробнее об этом в 9 уроке.
  - {% comment %} … {% endcomment %} – тег, используемый для комментирования.
  - {% debug %} – тег, выводящий большое количество информации для отладки, включая текущий контекст шаблона и все импортированные модули.
- Основные теги использования шаблонов Джанго:
  - {% include "foo/bar.html" %} – загрузить шаблон и отобразить его с контекстом текущего шаблона.
  - {% block block_name %} Текст по умолчанию. {% endblock %} – задает блок, который может быть переопределен расширяющими (наследующими) шаблонами. block_name – идентификатор, определяющий блок.
  - {% extends "base.html" %} – загружает указанный шаблон как родительский. При задании {% block block_name %} Текст для отображения. {% endblock %} – переопределит родительский block указанным текстом. Если какой-то блок не переопределен, то будет использован текст из родительского блока.



CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

- **Фильтры в шаблонах** – преобразуют переменные и аргументы тегов. Указываются через вертикальную черту, например {{ django|title }} – применит фильтр title к переменной django. Некоторые примеры фильтров:
    - default – если значение False (None, "", 0), то подставляется значение из default: {{ value|default:"nothing" }}
    - dictsort – принимает список словарей, и возвращает этот список отсортированным по ключу. Пример использования: {{ value:dictsort: "name" }}. Он также принимает значения вложенных ключей, указанных через точку, например {{ books|dictsort:"author.age" }. Также может сортировать вложенные списки по какому-то элементу, тогда в качестве ключа надо указывать индекс элемента, по которому проводится сортировка.
    - divisibleby – возвращает True, если значение делимо на аргумент без остатка. {{ value|divisibleby:"3" }}
    - Floatformat – форматирует Float, округляя его до указанного количества символов. При этом если количество указать с минусом, то в отсутствии дробной части не будет возвращать 0. Аргумент по умолчанию -1, т.е. округляет до 1 символа после запятой, не отображая 0, если дробная часть 0;
    - random – возвращает случайный элемент из списка;
    - stringformat – форматирует значение в соответствии с аргументом, аналогично printf-style форматированию, запись {{ value|stringformat:"E" }} эквивалентна записи "%e" % value, и если значение value у нас 10, то вывод будет 1.000000E+01
    - linenumbers – отображает текст с номерами строк
- Шаблонизатор **Jinja**. Синтаксис **Jinja2** сильно похож на Django-шаблонизатор, но при этом дает возможность использовать чистые Python выражения и поддерживает гибкую систему расширений. Также **Jinja2** поддерживается и другими фреймворками, а не только Django. Можно настроить Django так, чтобы в одном случае он использовал шаблонизатор Django, а для другого – Jinja2.
- Основное преимущество **Jinja** шаблонизатора – скорость, больше логики в шаблоны, универсальность (легче перенести на другие фреймворки).
- Основные преимущества Jinja: скорость, кастомные шаблонные теги, больше логики в шаблонах.

## Закрепление материала

- Для чего используются шаблоны в Django?
- Какие два backend-а для обработки шаблонов содержит в себе Django?
- Как осуществляется загрузка шаблонов в Django?
- Что такое Шаблон Django?
- Зачем нужен контекст в шаблоне?
- Какие 4 основные конструкции используются в шаблонах Django? Какой синтаксис используется для каждой конструкции?
- Какие есть основные преимущества у шаблонизатора Jinja?
- Можно ли подключить в один проект использование нескольких шаблонизаторов? Каким образом это делается?

## Дополнительное задание

Задание
У нас есть список:

latest_question_list = [{'id': 1, 'question_text': 'В чем смысл жизни?"}, {'id': 2, 'question_text': 'Что первично, дух или материя?"}, {'id': 3, 'question_text': 'Существует ли свобода воли?"}]



CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

Нужно составить шаблон, который проверяет значение переменной latest_question_list, и, если этот список не пустой, то выводит его элементы question в виде HTML списка <li>, где указывается ссылка на "/polls/question.id/", с текстом question.question_text. Если список пустой, то показывается строка "Список вопросов пуст".
Составить также представление, которое использует этот шаблон.

## Самостоятельная деятельность учащегося

Задание 1
Дан следующий список:

lets_do_it = [{'priority': 100, 'task': 'Составить список дел'}, {'priority': 150, 'task': 'Изучать Django'}, {'priority': 1, 'task': 'Подумать о смысле жизни'}]

Вывести его с помощью шаблона и фильтра в порядке убывания priority

Задание 2
Составить структуру сайта с помощью шаблонов так, чтобы у вас была главная страница с приветствием и ссылками на другие страницы сайта:

Добро пожаловать во вселенную звездных войн!
Выберите страницу:
Люк
Лея
Хан

и 3 страницы с общим заголовком «переход на главную», которые реализован с помощью одного общего шаблона, который они вместе наследуют, и уникальной для каждой страницы содержанием:

страница Люка:
Люк Скайуокер — один из главных персонажей вселенной «Звёздных войн», джедай, сын сенатора с Набу Падме Амидалы Наберри и рыцаря-джедая Энакина Скайуокера

страница Леи:
Лея Органа — дочь рыцаря-джедая Энакина Скайуокера и сенатора Падме Амидалы Наберри.

страница Хана:
Хан Соло — пилот космического корабля «Тысячелетний сокол», его бортмехаником и вторым пилотом является вуки по имени Чубакка.

Задание 3
Составить шаблон и использовать его в представлении, который бы отображал список словарей, хранящий информацию:

[{'name': 'Шаддам IV', 'surname': 'Коррино'}, {'name': 'Пол', 'surname': 'Атрейдес'}, {'name': 'Франклин', 'surname': 'Герберт'}]

с помощью тега for в виде списка:

```
<ol>
        <li> {name} {surname} </li>
</ol>
```



CyberBionic systematics
CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine
Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

## Рекомендуемые ресурсы

Официальная документация Django:
https://www.djangoproject.com/



CyberBionic Systematics ® 2020
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com